

uikit

一款轻量级、模块化的前端框架
可快速构建强大的web前端界面

目錄

UIkit 中文文档

开始使用

> 来源：开始使用

开发者

项目结构

深入挖掘UIkit的核心，了解关于它的一切。

GitHub

UIkit被托管在 GitHub 上，基于 MIT 许可协议。 欢迎使用它应用于个人或商业项目。想要获取 UIKit 所有的源文件，只需克隆 GitHub 上的资源库，或者直接下载最新的版本。

下载源代码

文件结构

UIkit 基于LESS创建，LESS是一款CSS预处理器，将 CSS 赋予了动态语言的特性，如变量，继承，运算，函数等。它将写入LESS中的代码编译成为CSS。它将写在Less里的代码编译成CSS。UIkit主要的文件结构如下：

文件夹	描述
`/docs`	包含你正在看的UIkit文档的绝大多数文件
`/src`	包含全部Less和JavaScript组件文件
`/tests`	包含所有组件的测试文件
`/themes`	在子文件夹中包含的所有额外提供的主题文件。
`/vendor`	包含UIkit所使用的jQuery和其他的外部库。

所有编译版和压缩版CSS文件、JS文件，以及Less和Sass文件都能在独立的 Bower UIkit repository 项目中找到。查看 Less 和 Sass 文件 获取更多信息。

自动地预处理

为了自动化处理Less文件编译成为CSS的过程，我们使用 Gulp，一款基于 Node.js 的前端构建工具，它监控你的工作目录。无论何时，只要你保存了有修改行为的源文件，Gulp都会自动地将所有文件编译成独立的CSS文件。

首先，你需要安装Node，并设置 `gulp` 作为一个全局的安装。最后，切换到UIkit目录，安装Node依赖模块（Node dependencies）。

```
npm install -g gulp

cd uikit
npm install
```

现在，你可以运行Gulp来创建和修改版本。UIkit的内置版本被放在 `/dist` 目录中。通过主题名称参数来只创建一个指定的主题。

```
gulp [-t themename]
```

你也可以设置Gulp监控你的工作目录，这样它便能在你每次保存时，自动地编译文件。通过主题名称参数来监控一个指定的主题以提升建设进度。

```
gulp watch [-t themename]
```

加载新的主题和样式到定制工具中。

```
gulp indexthemes
```

测试

UIkit为每个组件提供测试文件。其中每个页面都包含了组件的测试示例，并为你提供了一个关于盒模型所支持的所有可能性的总览。

UIkit 使用 BrowserSync 加速你的工作流程。运行 `gulp sync` ，你就能在制作自己的主题或者修改现有主题时，即时地查看改动，这不只是在浏览器中，还支持跨设备。你能在测试中切换所有的组件和主题，甚至切换到RTL模式（从右到左）

前往测试

Less & Sass 文件

一个涵盖包括 LESS 和 Sass 在内所有分发文件的独立 UIkit Bower 仓库。

UIkit 的伟大之处在于，你可以很轻松地将它的源文件集成到你的项目中，并能保障在你的自定义工作流能使用你最喜爱的CSS预处理器来构建资源（assets）。

UIkit 使用 Bower 包管理器来加载资源（assets）。因此 UIkit 自动生成分发文件到独立的 Bower UIkit repository。它包含所有来自 UIkit 及其组件的 CSS、Less、SCSS 和 Javascript文件。

获取最新的 UIkit 分发文件

1. 安装 Bower：`bower install uikit`
2. 或者，从 Github 获取包。

你会发现 Less 和 Sass 文件都存在，在它们各自独立的文件夹中：

```
/uikit
    less/
        uikit.less
        ...
    scss/
        uikit-mixins.scss
        uikit.scss
        ...
```

在你的项目中嵌入 Less

使用 Less 是真的直截了当。只需在 Less 文件的顶部引入 UIkit及其 附加组件，然后利用 UIkit 的钩子和覆写 UIkit 变量，就能开始定制了。

Example

```
@import "uikit/less/uikit.less";
@import "uikit/less/components/autocomplete.less";

// your custom code goes here, e.g. mixins, variables
```

在你的项目中嵌入 Sass

由于 Sass 解析代码的原因，它需要首先引入 `uikit-mixins.scss` ，然后添加自定义项（钩子、变量等），最后再加入 UIkit 和附加组件的文件。

Example

```
@import "uikit/scss/uikit-mixins.scss";

// 此处填入你的代码，比如： mixins, variables

@import "uikit/scss/uikit.scss";
@import "uikit/scss/components/autocomplete.scss";
```

创建主题

扩展UIkit并添加你自己的独一无二的主题。

如何开始

默认地，UIkit带有一个非常基础的主题。如果要修改它的样式，你不需要编辑任何核心框架文件。相反，你可以通过创建一个自定义主题来扩展UIKit。这样便允许你轻松地更新UIkit的核心文件。我们甚至提供了一些灵巧的主题以帮助你入门。要创建自己的主题，请按照下面步骤操作：（这里有一篇文章可以参考下：uikit框架开发前期配置及定制主题方法 ）

1. 创建主题目录

下载或克隆UIKit，安装 Node 和 Gulp 。最后，在这里创建你的主题文件夹 `/custom/THEME-NAME` 。如果 `/custom` 不存在，那就创建它！

注意 `/custom` 文件夹设置为 git ignore，这样可以防止你的自定义文件被推入到UIKit库中。如果你在使用克隆版 UIkit's git 仓库， 它将是一次很好的将 `/custom` 作为你自己的git仓库的实践。这样，你的主题文件的版本控制将不会受到UIkit文件的干扰。

2. 添加你的主题

在 `/custom/THEME-NAME` 文件夹中创建 `uikit.less` 文件，并添加 `@import "../../src/uikit.less";` 规则来访问核心框架中的所有Less文件并采用它的基础样式。好了，你可以从零开始添加你自己的主题了。

注意 为了能立即开始你的工作，我们提供了默认、渐变和扁平三个主题。所有默认文件都已经引入了，你同样可以找到所有主题文件以及与它们相关的钩子。所以你需要做的是复制文件夹到 `/custom` 目录中，并重命名。

3. 定制和测试

在你的 UIkit 文件夹中运行 gulp 的 `indexthemes` 指令。现在，刚刚新建的主题就将会自动出现在定制工具和测试文件中。

4. 构建你的主题

你可以在定制工具中生成你的主题的 CSS。如果你想使用 gulp 指令来做这事，运行 `gulp dist -t THEME-NAME` 就行了。生成的文件被放置在 `dist/` 文件夹中。

最佳主题实践

设计你的主题有多种不同的方法，我们推荐以下的工作流程。

1. 使用变量

首先要做的是自定义已声明的变量的值。你可以在核心框架的Less文件中找到所有组件的变量，在你的主题中重写它们。

Example

`/src/core/button/button.less`

```
// 默认值
@button-height: 30px;
```

`/custom/THEME-NAME/button.less`

```
// 新的值
@button-height: 35px;
```

2. 使用钩子

为了防止架空选择器，我们使用 Less 的混合（Mixins）方法与 UIkit 源码中预定义的选择器进行挂钩，并运用其附加属性。选择器不必在所有文件中重复填写，全局的修改变得更容易了。

Example

`/src/core/panel/panel.less`

```
// CSS 规则
.uk-panel {
    background: @background;

    // mixin 混合增加新的声明
    .hook-panel;
}
```

`/custom/THEME-NAME/panel.less`

```
// mixin 混合增加新的声明
.hook-panel() { color: #fff; }
```

3. 混杂的钩子/Miscellaneous hooks

如果没有变量也没有钩子可用，当然你也可以自行创建一个你自己的选择器。为此，请使用 *.hook-panel-misc* 钩子并将你的选择器写入其中。这样将会把你的新选择器整合到编译后的CSS文件的合适位置。

Example

`/custom/THEME-NAME/panel.less`

```
// misc mixin
.hook-panel-misc() {

    // 新的规则
    .uk-panel a { color: #f00; }
}
```

创建样式

拓展UIkit并新建属于你的独一无二的样式。

仅仅通过一组变量就可以改变现有主题的外观。你可以为每个主题创造无限种的样式。要创建你自己的样式，请按照下列步骤操作：

1. 创建样式目录

在这里创建你的样式文件夹 `/custom/THEME-NAME/styles/STYLE-NAME`

2. 添加你的样式

在 `/custom/THEME-NAME/styles/STYLE-NAME` 文件夹中创建一个空白的 *style.less* 文件。现在你可以手动添加新的变量声明或者使用 定制工具。当你完成时，点击 *Get Less* 按钮保存在你的样式文件到你的样式到文件夹中。

3. 定制和测试

在UIkit目录中运行gulp task *indexthemes* 使你刚才创建的样式可以在定制工具和测试页面中可见。

4. 持续定制

现在，你可以继续 自定义 你的样式了。

Customizer.json

这个文件用于定义你的主题外观中的控件数量。

每个主题都有其特定的 `customizer.json` 。它定义了哪些变量是默认显示的，哪些变量仅在高级模式中显示。这个文件分为 *controls* 和 *groups*两部分。

Controls/控件

控件（controls）用于定义如何在定制工具中显示变量。默认情况下，所有的变量都显示在输入框（input）元素中。你可以使用以下类型（type）来改变默认的输入框。

Type	描述
`"type": "color"`	将含有色彩数值的输入框转变为易用的色彩选择器。
`"type": "select"`	使用这种类型得到一个选择框，替代输入框。
`"type": "font"`	将会转变成带类似url的附加选项的选择框或选择框组。

Variables/变量

选择了输入框的类型后，你还须定义哪些变量会相互影响。你可以使用特定的单个变量或者使用通配符 `*` 来选择一组，而不是一个一个地单独设置每个变量。

特定变量

只影响 `@global-border` 和 `@global-light-border` 。

```
{
    "vars": [
        "@global-border",
        "@global-light-border"
    ]
}
```

通配符变量

影响所有以 `-color` 和 `-background` 结尾的变量。

```
{
    "vars": [
        "*-color",
        "*-background"
    ]
}
```

颜色选择器

通过设置 `"type": "color"` 将下面的示例中所有以 `-color` 或 `-background` 结尾的变量变成一个简单易用的颜色选择器，需要注意的是变量的值必须是一个颜色值。

Example

```
{"controls": [
    {
        "type": "color",
        "vars": [
            "*-color",
            "*-background"
        ]
    }
]}
```

选择元素

如果一个变量只能使用特定的值，你可以通过使用 `"type": "select"` 轻松地将输入元素变成一个选择元素，并加入可选择的选项。

Example

```
{"controls": [
    {
        "type": "select",
        "vars": [
            "*-weight"
        ],
        "options": [
            {"name": "Normal", "value": "normal"},
            {"name": "Bold", "value": "bold"}
        ]
    }
]}
```

字体

当涉及到字体时，你可以使用上面介绍的“选择元素”方法或通过设置 `"type": "font"` 来创建一个字体选择框，你可以添加一些值，比如一个字体 *url* 或者将你的字体分成几组。

Example

```
{"controls": [
    {
        "type": "font",
        "vars": [
            "*-font-family"
        ],
        "options": {
            "System Fonts": [
                {"name": "Arial", "value": "\"Helvetica Neue\", Hel
                {"name": "Consolas", "value": "Consolas, monospace,
            ],
            "Google Fonts": [
                {"name": "Abel", "value": "'Abel'", "url":"http://f
                {"name": "Asul", "value": "'Asul'", "url":"http://f
        }
    }
]}
```

Groups/组

组（groups）定义了哪些变量应该显示在定制器的侧边栏中。你可以把一些相关的变量组合到一起放在一个组中，显示在一个标题下，或者在 *Advanced Mode(*高级模式*)*中显示。

Example

这两个变量的设置将被默认显示在定制器的侧边栏中。

```
{"groups": [
    {
        "label": "Backgrounds",
        "vars": [
            "@global-background",
            "@global-dark-background"
        ]
    }
]}
```

在高级模式中显示导航栏组件的所有变量。

```
{"groups": [
    {
        "label": "Navbar",
        "advanced": true,
        "vars": [
            "@navbar-*"
        ]
    },
]}
```

JavaScript

使用 data 属性，将JavaScript应用于 UIkit 的组件。

你只需要在HTML元素中添加 `data-uk-*` 属性就能使用所有的UIKIT组件，而无需无需编写一行 JavaScript。这是在 UIkit 中使用任意组件时应当首先考虑的最佳做法。

Markup

```
<button data-uk-button>My Button</button>
```

当然你仍然可以通过使用 JQuery 的 API 来使用这些组件。

Markup

```
$(".button").uk("button");
```

AMD 支持

AMD (异步模块定义) 是一种定义 JavaScript 模块以及模块之间的依赖性的方式，因此他们可以通过异步加载的方式来使用。

用法

```
/* UIkit 核心的简单请求 */
require("path/to/uikit.js", function(UI){

    // UI 是UIkit的全局对象，又名 $.UIkit

});
```

自动加载 UIkit 及附加组件

```
/* 首先建立 require.js  */
requirejs.config({

    paths: {
        "uikit": 'path/to/uikit.js'
    },

    config: {
        "uikit": {
            "base": "path/to/uikit_dist_folder"
        }
    }

});

/* 现在，你可以自动加载UIkit和附加组件了，用逗号隔开。 */
require("uikit!notify,sortable", function(UI){

    // 访问已加载的附加组件: UI.notify, UI.sortable

});
```

覆写默认的组件设置

调整默认的组件设置是可行的，你可以看一下下面的例子：

用法

```
 // 覆写默认的工具提示设置
                           UIkit.on('beforeready.uk.dom', functior

                               $.extend(UIkit.components.tooltip.p
                                   pos: 'top',
                                   delay: 500,
                                   animation: true
                               });
                           });
```

监视**DOM**，通过诸如**AJAX**的方式自动初始化新添加的组件。

如果你希望通过JavaScript动态地将HTML标签注入到DOM中，只需要添加 `data-uk-observe` 属性到一个父元素中，就能自动地初始化UIkit的JavaScript组件。

用法

```
<div data-uk-observe>
                                <!-- 在此注入你的动态HTML -->
                            </div>
```

通过JavaScript监视元素

```
UIkit.domObserve('#element', function(element) { /* 适用于元素内部的D
```

基于可见性变化，检测显示事件/Check Display event on visibility change.

有时，一些组件，比如 网格 或 选项卡 被隐藏在标签中。这或许是与 切换器, 模态对话框 或者 下拉菜单 组合使用是发生的。一旦它们变得可见，他们需要被重新计算修改高度或者其他外形尺寸。

为此，添加 `data-uk-check-display` 属性到需要重新处理的元素中。现在，它们监听由容器组件触发的 `display.uk.check` 事件，比如切换器。带有 `data-uk-margin` , `data-uk-grid-margin` 和 `data-uk-grid-match` 属性的元素不需要添加这个属性，因为它此时是默认触发的。

Usage

```
<!-- 一个位于模态对话框、切换器或者下拉菜单中的元素 -->
                            <div id="myelement" data-uk-check-displ

                            <script>
                            $("#myelement").on('display.uk.check',
                               // 自定义的代码，用于调整显示的高度等
                            });
                            </script>
```

自定义前缀

创建自定义前缀的UIkit构建，是为了在同时使用多个版本的UIkit时，避免出现冲突。

随着我们不断地发行新版的UIkit，当越来越多使用UIkit构建的主题和扩展被加载到同一个页面上时，一些组件或者class会被改变或失效。当class在两个版本上作用不一样时，这可能引起冲突。

这就是为什么你可以自定义UIkit的前缀。这基本上会用你设置的前缀来取代通常的 `uk-` 。

通过 Less 设置前缀

如果你不熟悉 `gulp` 的用法，查看 自动预处理 了解如何创建一个UIkit构建。创建自定义前缀的UIkit版本，只需要在 `-p` 中使用你自己的前缀参数，即可使所有的class和JavaScript文件变成自定义前缀。

```
gulp -p myprefix
```

运行gulp后，你将在dist文件夹中找到你的自定义前缀构建。比如：

```
.myprefix-grid { ... }
                          ...
```

JavaScript 的无冲突模式

为了防止多个版本的自定义UIkit发生冲突，主需要在包含UIkit 之后调用 `noConflict` 方法:

```
var myUIkit = UIkit.noConflict();
```

核心组件

> 来源：核心组件

UIkit 全部组件一览。

UIKit提供了超过 30 个模块化并可扩展的组件，它们可以组合使用。根据用途及功能，我们把组件分了为不同的类型。

默认

这些组件通常用来实现HTML元素的跨浏览器标准化功能，并添加了一些非常基础的样式。

布局

充分运用我们完全响应式的流体网格系统和面板，常见的布局组成部分如博客文章和评论以及其他一些实用的效果类型。

导航

UIKit 提供了不同形式的导航组件，如导航栏和侧导航。使用面包屑或通过一个分页来翻阅文章。

元素

基本的HTML元素样式，如表格和表单。这些组件使用自有的类。它们不会因想到任何默认的元素样式。

常用

在这里你可以找到一些经常在内容中使用的组件，比如按钮，徽章，这种，动画和其他很多的组件。

JavaScript

这些组件主要依赖于 JavaScript 的淡出显示与隐藏内容，比如下拉菜单，模态对话框，抽屉和提示组件等。

断点

UIkit 包含一系列为各种不同的视图宽度实现响应式内容的类。下面的表格提供了一个关于一些可用的视图断点以及相应的设备概述。你可以通过 定制器 来调整所有的断点。

大小	断点	设备
Mini	小于 479px	手机纵向
Small	480px 到 767px	手机横向
Medium	768px 到 959px	平板电脑纵向
Large	960px 到 1199px	台式机或平板电脑横向
Xlarge	1200px 或以上	大屏幕设备

CSS 架构

为了避免与其他CSS框架冲突，所有的UIKit类均以 `uk-` 作为前缀进行命名。组件分为组件本身、子对象和调节器，其类名通常沿用组件名。

定义

对象	概述
组件	理论上一个网站经常用到一部分基于类的组件模块，比如按钮。这些都可重复使用和相互组合。
子对象	子对象都被放置在一个组件中用来增强其功能性，比如一个在提示框中的关闭按钮。
修饰	修饰类class 用来调整修改组件及其子对象的样式，比如按钮的颜色和大小。

默认

基础

提供所有HTML元素的默认样式。

这个组件定义了你的网页的基础视觉效果。通过为每个HTML元素设置默认的色彩、间距、字体尺寸和其他属性实现强大的版面效果。本页是一个关于如何使用基础HTML元素和UIkit如何对它们进行风格化的简要向导。

注意UIkit基本上利用了有名的 Normalize.css，但将它分散在了所有组件。只有必要的部分被采纳到基础组件中，以保证未预设的CSS样式尽可能的少。大部分关于CSS标准化的代码被移入了 表单, 按钮 和 表格 这些组件中。这样使得UIkit及其组件在与其他竞争中显得很厉害的样子，所以你不必担心你的项目在不同的浏览器中的一致性问题。

标题

使用 `<h1>` 到 `<h6>` 标签来定义你的标题。

Example

h1 标题 1

h2 标题 2

h3 标题 3

h4 标题 4

h5 标题 5

h6 标题 6

你可以添加 `.uk-h1` ， `.uk-h2` ， `.uk-h3` ， `.uk-h4` ， `.uk-h5` 或 `.uk-h6` 类来改变你的标题的大小，比如让 `h1` 看起来像一个 `h3` 。

段落

通过可以自定义设置的变量，可对全局的字体大小、行间距、边距等常规元素进行调整。段落和其他块元素也继承这些变量值。

文本级语义

下面的列表为你提供了最常用的文本层次语义，以及如何利用它们的简短概述。

标签	描述
`<a>`	将文本转换为超文本使用 a 标签。
`<em>`	强调文本使用 *em* 标签。
`<strong>`	强调任何额外的更重要文本使用 **strong** 标签。
`<code>`	定义内联代码片段使用 `code 标签` 。
`<del>`	标记文档中已被修改或删除的文本使用 ~~del 标签~~。
`<ins>`	标记文档中插入的文本使用 <ins>ins 标签</ins>。
`<mark>`	高亮显示文本使用 <mark>mark 标签</mark>，它没有任何语义含义。
`<q>`	定义引入的语录在 <q>q 标签 <q>里面使用</q> q 标签</q>。
`<abbr>`	定义一个标题的缩写使用 <abbr title="Abbreviation Element">abbr 标签</abbr>。
`<dfn>`	定义一个项目术语名称使用 <dfn title="Defines a definition term">dfn 标签</dfn>。
`<small>`	缩小显示不重要的文本使用 small 标签。

水平线

使用 `<hr>` 标签创建一条水平线。

Example

引用文字

在文档中从其他来源引用多行内容时，使用 `<blockquote>` 标签。

Example

> *Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam.*
>
> *Someone famous*

Markup

```
<blockquote>
    <p>Quotation</p>
    <small>Source</small>
</blockquote>
```

代码块

对于多行代码，使用 `<pre>` 元素定义预格式化文本。它能创建提供保留空格、制表符和换行符的新文本块。内部嵌套 `<code>` 标签来定义代码块。

重要 为保证正确地渲染，必须确保避开代码中的任何尖括号。

Example

```
<pre>
    <code>...</code>
</pre>
```

注意 你也可以添加 效果组件 中介绍到的 `.uk-scrollable-text` 类，这将设置最大高度为300px，并提供Y轴滚动条。

列表

创建一个无序列表使用 <ul> 标签或使用 <ol> 标签创建有序列表。 元素定义列表项。

Example

- Item 1
- Item 2
    - Item 1
    - Item 2
        - Item 1
        - Item 2
- Item 3
- Item 4

1. Item 1
2. Item 2
    1. Item 1
    2. Item 2
        1. Item 1
        2. Item 2
3. Item 3
4. Item 4

Markup

```
<ul>
    <li>...</li>
    <li>...
        <ul>
            <li>...</li>
        </ul>
    </li>
</ul>
```

```
<ol>
    <li>...</li>
    <li>...
        <ol>
            <li>...</li>
        </ol>
    </li>
</ol>
```

描述列表

创建一个描述列表需要使用 <dl> 标签。使用 <dt> 定义项目，使用 <dd> 定义描述内容。

Example

Description lists
A description list defines terms and their corresponding descriptions.
This is a term
This is a description.
This is a term
This is a description.

Markup

```
<dl>
    <dt>...</dt>
    <dd>...</dd>
</dl>
```

响应式图片

UIKit 中的所有图片默认都是响应的。如果布局变窄，图片会调整它们的大小并保持自身的比例。

Example



注意 若要避免响应式行为并保持图像的原始尺寸，可以为单独的图片添加 `.uk-img-preserve` 类。如果你具有多张图片，也可以将这个类添加到父容器中。谷歌地图不需要这样的设置。

打印

优化您的页面用于经济的打印。

这个组件基于 HTML5 Boilerplate 的打印样式。它将去掉页面的背景颜色并改变字体颜色为黑色以节省打印机墨水。图片将被按比例缩小以适合页面，并且锚文本将带下划线以区别于常规文本。

修改

对于 UIkit，我们移除了链接的 `href` 及 `title` 相关的规则。

布局类组件

网格

创建一个完全响应式并可以嵌套的流动网格布局。

UIKit中的网格系统遵循移动优先的方式并且最多可容纳10个网格列。它使用网格内预定义的类对每个单元格的列宽进行了定义。另外，还可以把网格与 Flex 组件 中的类组合使用，虽然它只能在现代浏览器中正常运行。

用法

向一个父元素添加 `.uk-grid` 类来创建网格容器。对子元素添加一个 `.uk-width-*` 类来限定单元格的宽度。网格支持1、2、3、4、5、6和10个单元划分。下面的列表为你提供了可以应用到单元的 `uk-width-*` 类的概述。

Class	描述
`.uk-width-1-1`	填满可见宽度的100%。
`.uk-width-1-2`	把网格等分为两半。
`.uk-width-1-3` to `.uk-width-2-3`	将网格划分成三分之一。
`.uk-width-1-4` to `.uk-width-3-4`	将网格划分成四分之一。
`.uk-width-1-5` to `.uk-width-4-5`	将网格划分成五分之一。
`.uk-width-1-6` to `.uk-width-5-6`	将网格划分成六分之一。
`.uk-width-1-10` to `.uk-width-9-10`	将网格划分成十分之一。

我们特意为每组单元建立了一种冗余，所以在实际应用时， `.uk-width-5-10` 类与 `.uk-width-1-2` 类的实际效果都是一样的。

Example

仔细看看下面网格的例子，这可以使你了解所有 `.uk-width-*` 类的基本使用方法。



注意 网格并没有相关联的CSS样式。在示例中，我们使用了 面板组件。

Markup

下面是关于如何使用默认的两列网格代码简单示例：

```
<div class="uk-grid">
    <div class="uk-width-1-2">...</div>
    <div class="uk-width-1-2">...</div>
</div>
```

响应式宽度

UIKit中提供了一些非常有用的响应式宽度的类。基本上它们的使用方法就像通常宽度的类一样，但是它们带有前缀，这样它们只在特定的断点来产生效果。这些类可以结合 效果组件 中的可见性类来使用。这对于调整不同尺寸设备的布局和内容是非常重要的。

Class	描述
`.uk-width-*`	对于任何宽度的设备，网格都保持列并排。
`.uk-width-small-*`	影响宽度在 *480px* 以上的设备。网格列将在较小的视口中堆叠。
`.uk-width-medium-*`	影响宽度在 *768px* 以上的设备。网格列将在较小的视口中堆叠。
`.uk-width-large-*`	影响宽度在 *960px* 以上的设备。网格列将在较小的视口中堆叠。

重要 若要设置网格堆叠列上下间的边距，只需添加 `data-uk-grid-margin` 属性。

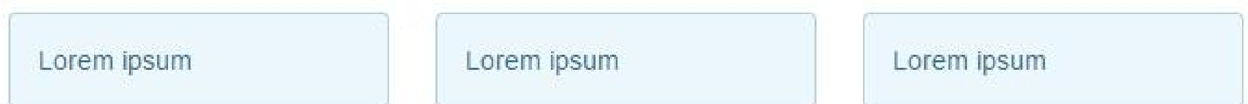
Example



网格排水沟/Grid gutter

好吧，其实是网格之间的间隔。网格会自动在列之间创建一个水平间距，以及在连续的两个网格间创建一个垂直方向的间隔。默认情况下，网格间隔在大屏幕上看起来会比较宽。

Example



中等排水沟/Medium gutter

在网格之间应用中等的间隔，只需要添加 `.uk-grid-medium` 类。

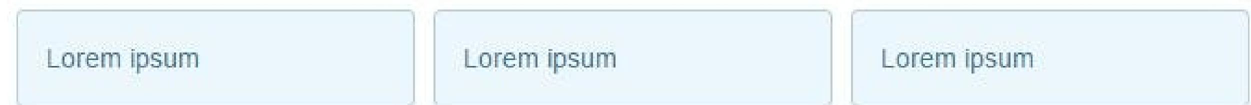
Example



小点的排水沟/Small gutter

在网格之间应用较小的间隔，只需要添加 `.uk-grid-small` 类。

Example



消除排水沟/Collapse gutter

完全地去掉间隔，只需要添加 `.uk-grid-collapse` 类。

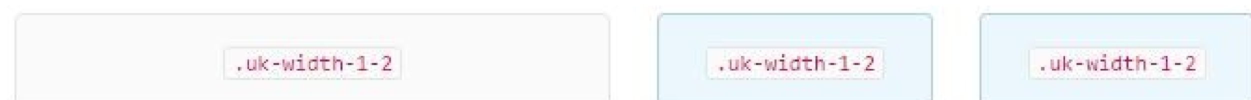
Example



网格的嵌套

使用嵌套网格你可以轻松地扩展你的网格布局。

Example



Markup

```
<div class="uk-grid">
    <div class="uk-width-1-2">...</div>
    <div class="uk-width-1-2">
        <div class="uk-grid">
            <div class="uk-width-1-2">...</div>
            <div class="uk-width-1-2">...</div>
        </div>
    </div>
</div>
```

居中网格

添加 效果组件 中的 `.uk-container-center` 类来让一个网格居中显示。

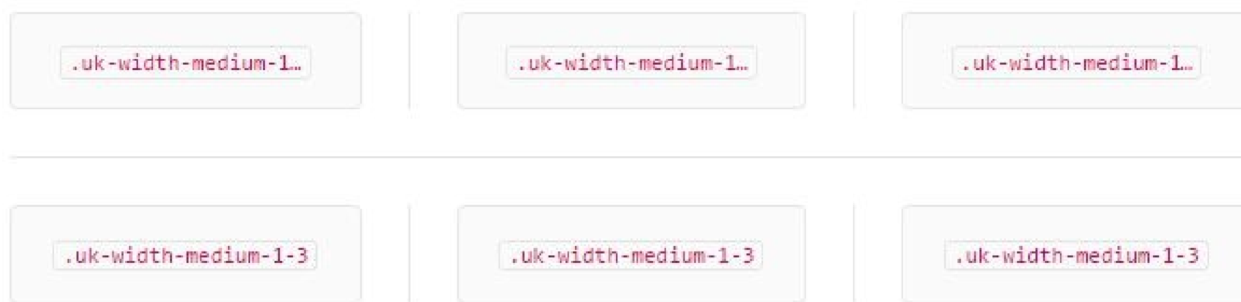
Example



网格分隔线

添加 `.uk-grid-divider` 类用线条分隔网格列。要使用水平线分隔网格，只需在 <hr> 或 <div> 元素添加这个类。

Example



Markup

```
<div class="uk-grid uk-grid-divider">...</div>
<hr class="uk-grid-divider">
<div class="uk-grid uk-grid-divider">...</div>
```

注意 水平分隔线不能用在含有 `uk-push-*` 类或 `uk-pull-*` 类的网格中。

源码排序

可以更改列的显示顺序，使其在源代码中保持一个特定的列顺序。添加 `.uk-push-*` 类，向右移动列。添加 `.uk-pull-*` 类，向左移动列。这可以使你进行类似翻转移动设备或改变窗口宽度时改变列的显示顺序。这些类还可以用于偏移列，在列之间建立额外的空隙。

源码排序对SEO和响应式设计是非常有用的，因为在狭窄的视口中网格会根据标签的源代码顺序来显示。

注意 此功能只能结合 `.uk-width-medium-*` 类来使用。

Example



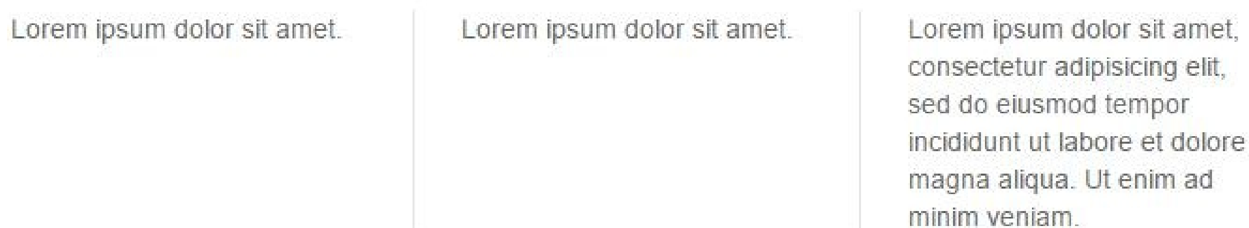
Markup

```
<div class="uk-grid">
    <div class="uk-width-medium-1-2 uk-push-1-2">...</div>
    <div class="uk-width-medium-1-2 uk-pull-1-2">...</div>
</div>
```

匹配列的高度

要匹配网格列的高度，只需要在你的网格添加 `data-uk-grid-match` 属性。如果你的网格包含多个row，只有同一个row中的网格列会被匹配。只需要使用 `data-uk-grid-match="{row: false}"` 属性就能跨越row的隔阂来匹配网格列的高度。

Example



Markup

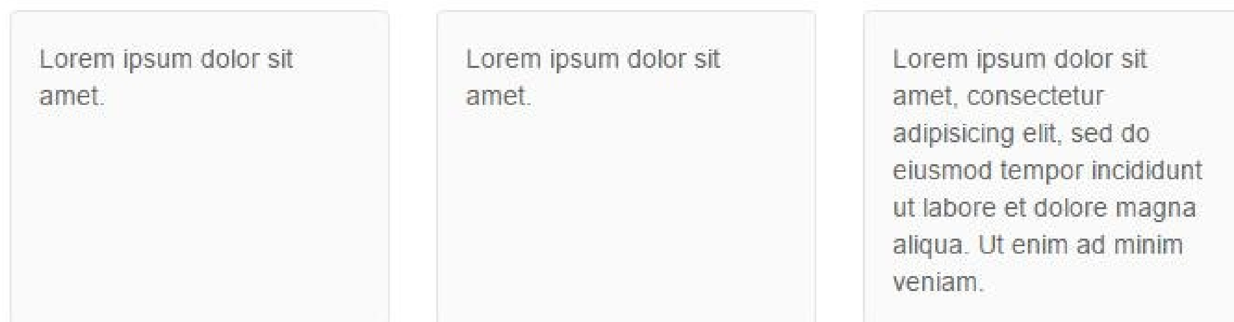
```
<div class="uk-grid" data-uk-grid-match>...</div>
```

注意 如果网格列的宽度延伸到100%，它们的高度将不再匹配。这样做是有道理的，比如说，当它们在一个较窄的视口中堆叠时。

匹配面板的高度

如果想要匹配一个网格中多个面板的高度，只需要添加 `.uk-grid-match` 类。在使用data属性时，必须添加 `{target:'.uk-panel'}` 选择器。

Example



Markup

```
<div class="uk-grid uk-grid-match" data-uk-grid-match="{target:'.uk
    <div class="uk-width-medium-1-3">
        <div class="uk-panel">...</div>
    </div>
</div>
```

包装多个行（row）

你也可以创建一个网格，使其包含任意多个列，这些列会自动地换到下一行。只需添加 `data-uk-grid-margin` 属性，就能创建网格行（row）之间的margin。通常这样的布局使用了 `<ul>` 元素。

Example



注意 你也可以对网格列的宽度进行自定义。只需添加 `.uk-width` 类，并使用内联样式定义宽度。下面的例子对宽度使用了固定的像素值，对于图片你也可以这样做。



Markup

```
<ul class="uk-grid" data-uk-grid-margin>

    <!-- 这些元素使用百分比宽度 -->
    <li class="uk-width-medium-1-5">...</li>
    <li class="uk-width-medium-3-10">...</li>

    <!-- 这些元素使用像素值宽度 -->
    <li class="uk-width" style="width: 100px;">...</li>
    <li class="uk-width" style="width: 150px;">...</li>

</ul>
```

均匀的网格列

要创建一个子元素的宽度都是均匀分割的网格，你不必为网格中的每个列表元素里添加同样的类。只需要添加一个 `.uk-grid-width-*` 类到网格本身即可。

Class	描述
`.uk-grid-width-1-2`	将网格均匀分为两份
`.uk-grid-width-1-3`	将网格均匀分为三份
`.uk-grid-width-1-4`	将网格均匀分为四份
`.uk-grid-width-1-5`	将网格均匀分为五份
`.uk-grid-width-1-6`	将网格均匀分为六份
`.uk-grid-width-1-10`	将网格均匀分为十份

Example



Markup

```
<ul class="uk-grid uk-grid-width-1-5">
    <li>...</li>
    <li>...</li>
</ul>
```

响应式宽度

UIkit同样提供了网格的响应式宽度类。可以用它保持均匀尺寸的网格列，无论设备宽度如何。

Class	描述
`.uk-grid-width-*`	影响所有设备宽度
`.uk-grid-width-small-*`	影响 *480px* 以上的设备宽度。
`.uk-grid-width-medium-*`	影响 *768px* 以上的设备宽度。
`.uk-grid-width-large-*`	影响 *960px* 以上的设备宽度。
`.uk-grid-width-xlarge-*`	影响 *1220px* 以上的设备宽度。

Example



Markup

```
<ul class="uk-grid uk-grid-width-1-2 uk-grid-width-medium-1-3 uk-gr
    <li>...</li>
    <li>...</li>
</ul>
```

面板/Panel

创建拥有不同样式的布局盒。

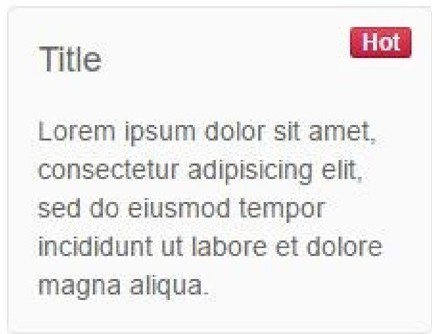
UIKit使用面板勾勒网页内容中的某些部分，它可以拥有不同的样式。通常，面板被布置在网格组件的网格列中。

用法

面板组件由面板本身，面板标题和面板徽章组成。为了防止产生多余的空白，面板内容顶部和底部的maigin都被移除了。

Class	描述
`.uk-panel`	为 `<div>` 元素添加这个类，定义面板组件。
`.uk-panel-title`	为标题元素添加这个类创建面板标题。
`.uk-panel-badge`	为 `<div>` 元素添加这个类创建一个面板的徽章。风格化徽章样式最简单的方法，就是通过加入徽章组件中的修饰符类。

Example



注意默认情况下，面板是空白的也没有样式。因此，为面板添加修饰符类的样式是很重要的。在例子中，我们使用了 `.uk-panel-box` 类。

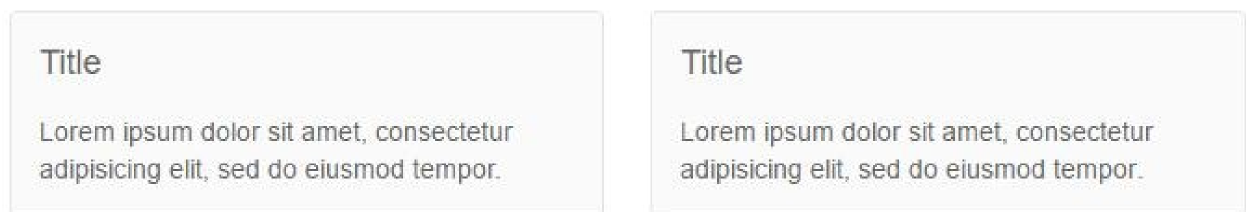
Markup

```
<div class="uk-panel">
    <div class="uk-panel-badge uk-badge">...</div>
    <h3 class="uk-panel-title">...</h3>
    ...
</div>
```

网格中的面板Panels in a grid

这是一个关于如何在网格组件中使用面板的简单例子。两个面板都使用了 `.uk-width-medium-1-2` 类。

Example



Markup

```
<div class="uk-grid">
    <div class="uk-width-medium-1-2">
        <div class="uk-panel">...</div>
    </div>
    <div class="uk-width-medium-1-2">
        <div class="uk-panel">...</div>
    </div>
</div>
```

修饰

使用修饰类为面板添加一个特定的样式是很有必要的。UIkit含有多种面板修饰，你也可以自己创建。

面板框

添加 `.uk-panel-box` 类来创建一个可见的面板框。这是默认的面板修饰样式。

Example



Markup

```
<div class="uk-panel uk-panel-box">...</div>
```

注意 为面板创建鼠标经过效果，只需添加 `.uk-panel-box-hover` 类。这将在使用锚文本时带来方便。

醒目的面板盒/Panel box primary

添加 `.uk-panel-box-primary` 类来修饰面板框，并以不同的颜色使其显得突出。

Example



Markup

```
<div class="uk-panel uk-panel-box uk-panel-box-primary">...</div>
```

注意 为面板创建鼠标经过效果，只需添加 `.uk-panel-box-primary-hover` 类。这将在使用锚文本时带来方便。

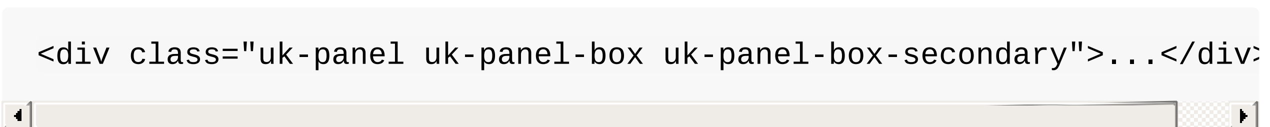
次要的面板盒/Panel box secondary

为面板框添加 `.uk-panel-box-secondary` 类，给它一个白色的背景。

Example



Markup



```
<div class="uk-panel uk-panel-box uk-panel-box-secondary">...</div>
```

注意 为面板创建鼠标经过效果，只需添加 `.uk-panel-box-secondary-hover` 类。这将在使用锚文本时带来方便。

鼠标经过面板/Panel hover

添加 `.uk-panel-hover` 类为面板天鼠标经过效果，这将为用作锚的面板带来便利。

Example



Markup



```
<a class="uk-panel uk-panel-hover" href="">...</a>
```

面板标题/Panel header

添加 `.uk-panel-header` 类，用一条水平线分隔面板的标题和内容。

Example



Markup

```
<div class="uk-panel uk-panel-header">...</div>
```

面板间距

添加 `.uk-panel-space` 添加类来增加面板四周的间距。

Example



Markup

```
<div class="uk-panel uk-panel-space">...</div>
```

面板分隔线

添加 `.uk-panel-divider` 类将垂直堆叠的面板用水平线进行分隔。

Example

Title

Lorem ipsum dolor sit amet, consectetur adipisicing elit. `.uk-panel-divider`

Title

Lorem ipsum dolor sit amet, consectetur adipisicing elit. `.uk-panel-divider`

Title

Lorem ipsum dolor sit amet, consectetur adipisicing elit. `.uk-panel-divider`

Markup

```
<div class="uk-grid">
    <div class="uk-width-medium-1-2">
        <div class="uk-panel uk-panel-divider">...</div>
        <div class="uk-panel uk-panel-divider">...</div>
        <div class="uk-panel uk-panel-divider">...</div>
    </div>
</div>
```

注意使用网格组件中的 `.uk-grid-divider` 类分隔网格列。

带图片预览的面板框

为了在一个面板内显示不带有任何边框的图片，仅需添加 `.uk-panel-teaser` 类至该面板内部的 `<div>` 元素。

Example



Markup

```
<div class="uk-panel uk-panel-box">
    <div class="uk-panel-teaser">
        <img src="" alt="">
    </div>
</div>
```

带图标的面板

在面板标题中为 `<i>` 或 `<span>` 元素添加一个 `.uk-icon-*` 类，可以轻松地将 图标组件中的图标应用在面板中。

Example



Markup

```
<div class="uk-panel">
    <h3 class="uk-panel-title"><i class="uk-icon-*"></i>...</h3>
</div>
```

块/Block

通过将内容片段打包成拥有不同样式的块来分割内容。

用法

只需要为容器元素添加 `.uk-block` 类，就能使用这个组件了。

Example

Block

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna.

Markup

```
<div class="uk-block">...</div>
```

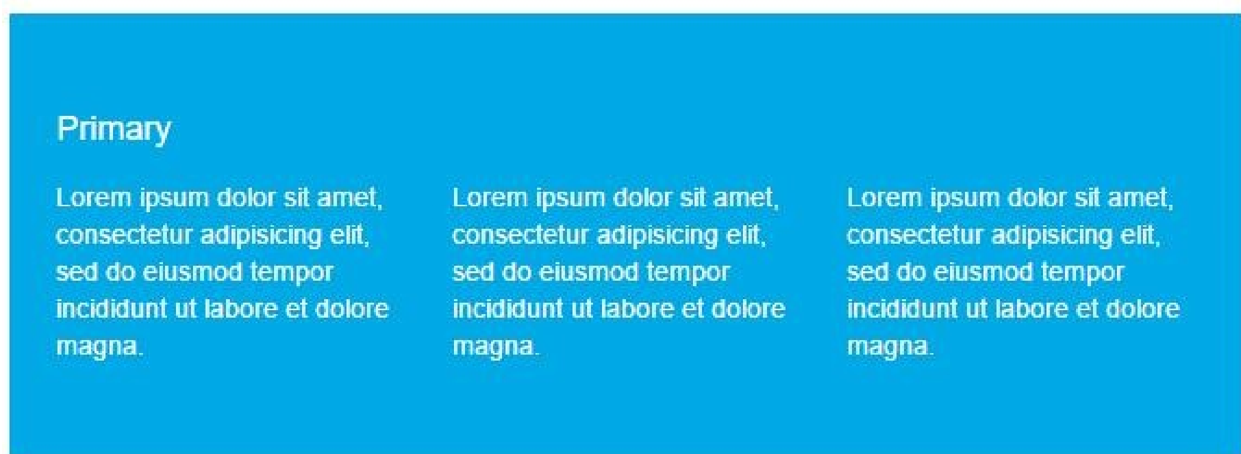
修饰

使用不同的背景颜色和 padding ，添加以下类中的一个就行了。当两个连续的块拥有相同的背景色时，padding 会自动被分开。

Class	描述
`.uk-block-default`	添加默认的背景色彩，通常是白色或类似的颜色。
`.uk-block-muted`	添加亮背景色。
`.uk-block-primary`	添加表示着重的背景色。
`.uk-block-secondary`	添加另一种背景色，通常是暗色。

注意 为了在有色的块中恰到好处地显示色彩、border和其他元素，你可能需要使用对比度组件中的 `.uk-contrast` 类。

Example



Markup

```
<div class="uk-block uk-block-primary">...</div>
```

Padding

为块添加较大的 padding ，只需添加一个 `.uk-block-large` 类。你还可以使用效果组件中的 `.uk-padding-*` 类，来设置块中 padding 。

Example

Large

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna.

Markup

```
<div class="uk-block uk-block-large">...</div>
```

文章

在你页面中创建文章。

用法

文章组件由文章本身、标题、元数据（例如发布时间、所属栏目、作者等）、起始段落（摘要）和间隔组成。

Class	描述
`.uk-article`	添加这个类来定义文章组件。通常你需要使用 `<article>` 元素。
`.uk-article-title`	对一个标题添加这个类，来创建文章标题。通常会用在 `<h1>` 元素中。
`.uk-article-meta`	向包含文章元数据的段落添加这个类。
`.uk-article-lead`	为文章起始段落添加这个类，使其显得突出。
`.uk-article-divider`	对 `<hr>` 元素添加这个类，建立一个间隔使文章各部分分开。

示例

文章标题-洛神赋

由 曹植 撰写于 2012年12月12日 | 发表在 博客

黄初三年，余朝京师，还济洛川。古人有言，斯水之神，名曰宓妃。感宋玉对楚王神女之事，遂作斯赋，其词曰：

黄初三年，余朝京师，还济洛川。古人有言，斯水之神，名曰宓妃。感宋玉对楚王神女之事，遂作斯赋，其词曰：

余从京域，言归东藩，背伊阙 ，越轘辕，经通谷，陵景山。日既西倾，车殆马烦。尔乃税驾乎蘅皋，秣驷乎芝田，容与乎阳林，流眄乎洛川。于是精移神骇，忽焉思散。俯则未察，仰以殊观。睹一丽人，于岩之畔。乃援御者而告之曰："尔有觌于彼者乎？彼何人斯，若此之艳也！"御者对曰："臣闻河洛之神，名曰宓妃。然则君王所见，无乃是乎？其状若何，臣愿闻之。"

继续阅读

Code

```
<article class="uk-article">
    <h1 class="uk-article-title">...</h1>
    <p class="uk-article-meta">...</p>
    <p class="uk-article-lead">...</p>
    ...
    <hr class="uk-article-divider">
    ...
</article>
```

评论/Comment

创建评论，比如文章的评论。

用法

评论组件由包含头像、标题和元数据的评论header以及评论主体组成。

Class	描述
`.uk-comment`	添加这个类定义评论组件。
`.uk-comment-header`	添加这个类创建评论header。
`.uk-comment-avatar`	添加到 <img> 元素创建评论者的头像。
`.uk-comment-title`	添加这个类到一个标题元素中，创建评论的标题。
`.uk-comment-meta`	为段落添加这个类，创建评论的元数据。
`.uk-comment-body`	为 <div> 元素添加这个类，创建评论主体。

示例

作者

12 天前 | 传说中的大水比 | #

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua.

Code

```
<article class="uk-comment">
    <header class="uk-comment-header">
        <img class="uk-comment-avatar" src="" alt="">
        <h4 class="uk-comment-title">...</h4>
        <div class="uk-comment-meta">...</div>
    </header>
    <div class="uk-comment-body">...</div>
</article>
```

评论列表

添加 `.uk-comment-list` 类到一个 `<ul>` 元素中，创建一个评论列表。你可以在评论列表中嵌套任意数量的 `<ul>` 元素。

示例



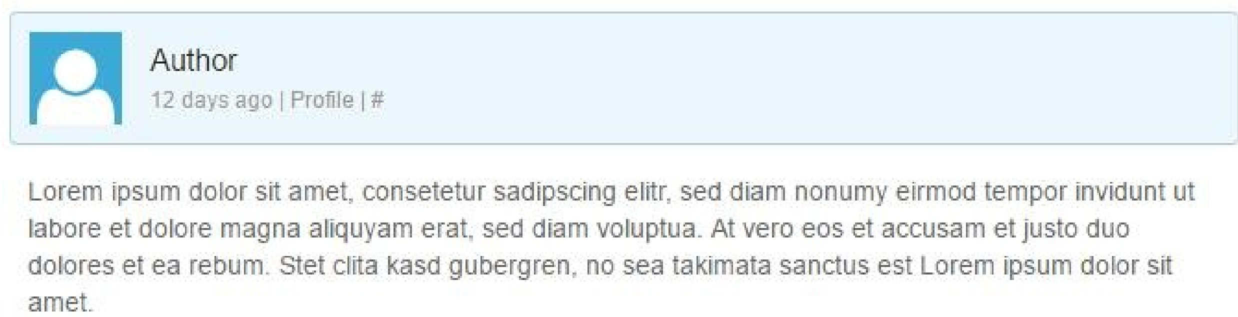
Code

```
<ul class="uk-comment-list">
    <li>
        <article class="uk-comment">...</article>
        <ul>
            <li><article class="uk-comment">...</article></li>
        </ul>
    </li>
    <li><article class="uk-comment">...</article></li>
</ul>
```

色彩调整

添加 `.uk-comment-primary` 类可以将评论样式变得不同，例如把管理员的评论设置为高亮。

示例



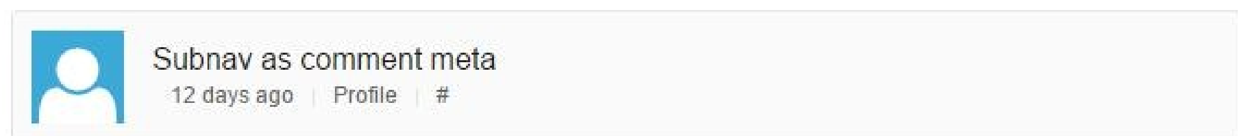
Code

```
<article class="uk-comment uk-comment-primary">...</article>
```

评论中的二级导航

使用二级导航组件 以二级导航栏的形式展示评论的元数据。

示例



Code

```
<ul class="uk-comment-meta uk-subnav uk-subnav-line">
    <li>...</li>
</ul>
```

效果/Utility

一些实用的效果类集合，它们可以用来风格化你的网页内容。

容器

添加 `.uk-container` 类到一个块元素中，为其设置一个最大宽度并将网页的主要内容包装在其中。对于大屏幕采用不同的最大宽度。

居中

想要将容器居中，使用 `.uk-container-center` 类。对于其它的块元素，你需要设定一个宽度。

示例



Code

```
<div class="uk-width-medium-1-2 uk-container-center">...</div>
```

浮动与清除浮动

浮动是创建各式布局的基础。但浮动需要清除，否则在最坏的情况下需，你会得到一个杂乱无章的页面。下面的类能帮助你创建基础的布局。

Class	概述
`.uk-float-left`	浮动元素为左对齐。
`.uk-float-right`	浮动元素为右对齐。
`.uk-clearfix`	向父容器添加这个类来清除浮动。

Code

```
<div class="uk-clearfix">
    <div class="uk-float-right">...</div>
    <div class="uk-float-left">...</div>
</div>
```

单独的用来格式化上下文的块元素/New block formatting context

代替使用 `.uk-clearfix` 类，你可以单独创建一个块元素来清除上下文的浮动。基于你的设置，你可以对其进行评估哪种方式更加适合。

Class	概述
`.uk-nbfc`	设置 overflow 为 hidden 来创建一个单独的用来格式化上下文的块元素。
`.uk-nbfc-alt`	设置 display 为 table-cell 来创建一个单独的用来格式化上下文的块元素。

图片与对象的对齐

有间距地将图片与其他元素（比如文本）对齐。

Class	概述
`.uk-align-left`	向左浮动元素，并创建右侧及底部的间距。
`.uk-align-right`	向右浮动元素，并创建左侧及底部的间距。
`.uk-align-medium-left`	仅影响宽度 *768px* 及以上的设备。
`.uk-align-medium-right`	仅影响宽度 *768px* 及以上的设备。
`.uk-align-center`	居中对齐元素，并创建底部间距。

示例

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.



Code

```
<p class="uk-clearfix">
    <img class="uk-align-medium-right" src="" alt="">
    ...
</p>
```

垂直对齐

将对象垂直对齐，你必须为需要对齐的对象创建一个父容器。

Class	概述
`.uk-vertical-align`	向父容器添加这个类。这个容器需要被设定一个高度。
`.uk-vertical-align-middle`	向子元素添加这个类，使内容居中对齐。
`.uk-vertical-align-bottom`	向子元素添加这个类，使内容与底部对齐。
`.uk-height-1-1`	这个辅助类用来赋予100%的高度。

示例

Lorem ipsum dolor sit amet, consectetur adipisicing elit.

Lorem ipsum dolor sit amet, consectetur adipisicing elit.

Code

```
<div class="uk-vertical-align">
    <div class="uk-vertical-align-middle">
    ...
    </div>
</div>
```

```
<div class="uk-vertical-align">
    <div class="uk-vertical-align-bottom">
    ...
    </div>
</div>
```

注意 需要对齐的元素需要有一个宽度或最大宽度，它的宽度必须比父容器宽度小。

居中整个页面

如果你想将 `<html>` 和 `<body>` 元素扩展到整个页面的高度， `.uk-height-1-1` 类便派上了用场。创建错误页面时，这是非常有用的。

Code

```
<html class="uk-height-1-1">
    ...
    <body class="uk-height-1-1">
        <div class="uk-vertical-align">
            <div class="uk-vertical-align-middle">...</div>
        </div>
    </body>
</html>
```

水平居中

水平居中你的网页内容，添加 `.uk-text-center` 类到父容器并添加 `.uk-container-center` 类到子元素。这是由于响应式特性而必须这样的。

视窗高度

添加 `.uk-height-viewport` 类，就可以创建一个填充整个视窗高度的容器，例如用于全屏图像或视频。

元素的定位

UIkit提供一系列的类去定位你的内容，而无须更改你自己的CSS。

Class	Description
`.uk-position-top`	将元素绝对定位在顶部
`.uk-position-top-left`	将元素绝对定位在左侧顶部
`.uk-position-top-right`	将元素绝对定位在右侧顶部
`.uk-position-bottom`	将元素绝对定位在底部
`.uk-position-bottom-left`	将元素绝对定位在左侧底部
`.uk-position-bottom-right`	将元素绝对定位在右侧底部
`.uk-position-cover`	为元素添加绝对定位，并将其扩展覆盖其父元素
`.uk-position-relative`	为元素添加relative定位方法
`.uk-position-z-index`	为元素添加数值为1的 *z-index* 属性

响应式对象

在UIkit中，图片会默认地自适应它的父容器。如果你想将响应式特性应用于媒体元素，比如视频对象，只需要添加下面的类中的一个。

Class	描述
`.uk-responsive-width`	根据父容器的宽度调整对象的宽度，保持对象原始的宽高比。
`.uk-responsive-height`	根据父容器的高度调整对象的高度，保持对象原始的宽高比。

NOTE 同样，可以添加 `.uk-responsive-width` 类名到 iframe ，此iframe需要有预设的width和height属性。

Example 宽度



Example 高度



Markup

```
<video controls class="uk-responsive-width"></video>

<img class="uk-responsive-height" src="" alt="">
```

行内的 SVG

SVG 或可缩放的矢量图形真的很棒，比如作为LOGO使用时，无论是缩放还是用于动画，它都很清晰。要使用 CSS 来控制 SVG，只需要为图片标签添加 `data-uk-svg` 属性。这可以将 SVG 转变行内元素，并具有所有属性，包括 ID、class、width、height 等等，你可以通过 CSS 方便地控制它。

Markup

```
<img src="my-image.svg" data-uk-svg>
```

外边距 Margin

添加一个下面的类为块元素添加外边距。

Class	描述
`.uk-margin`	为一个段落添加相同的顶部和底部外边距。
`.uk-margin-top`	添加顶部外边距。
`.uk-margin-bottom`	添加底部外边距。
`.uk-margin-left`	添加左侧外边距。
`.uk-margin-right`	添加右侧外边距。

较大的外边距

使用一个下面的类来为块元素添加较大的外边距。

Class	描述Description
`.uk-margin-large`	为一个段落添加较大的顶部和底部外边距。
`.uk-margin-large-top`	添加较大的顶部外边距。
`.uk-margin-large-bottom`	添加较大的底部外边距。
`.uk-margin-large-left`	添加较大的左侧外边距。
`.uk-margin-large-right`	添加较大的右侧外边距。

较小的外边距

使用一个下面的类来为块元素添加较小的外边距。

Class	描述
`.uk-margin-small`	为一个段落添加较小的顶部和底部外边距。
`.uk-margin-small-top`	添加较小的顶部外边距。
`.uk-margin-small-bottom`	添加较小的底部外边距。
`.uk-margin-small-left`	添加较小的左侧外边距。
`.uk-margin-small-right`	添加较小的右侧外边距。

移除外边距

添加一个下面的类来移除块元素的外边距。

Class	描述
`.uk-margin-remove`	移除全部外边距。
`.uk-margin-top-remove`	移除顶部外边距。
`.uk-margin-bottom-remove`	移除底部外边距。

自动外边距Auto margin

为堆叠的多个元素间距，例如，在较小的视口中堆叠显示多个并列的内联元素时，只需要添加 `data-uk-margin` 属性到它们的父元素，即可自动为子元素添加 the `.uk-margin-small-top` 。

示例



Code

```
<p data-uk-margin>
    <button class="uk-button">...</button>
    <button class="uk-button">...</button>
</p>
```

注意 默认情况下，data属性为堆叠的元素添加 `.uk-margin-small-top` 类。如果需要添加更大的margin，只需要添加 `{cls:'.uk-margin-top'}` 选项就行。

Padding

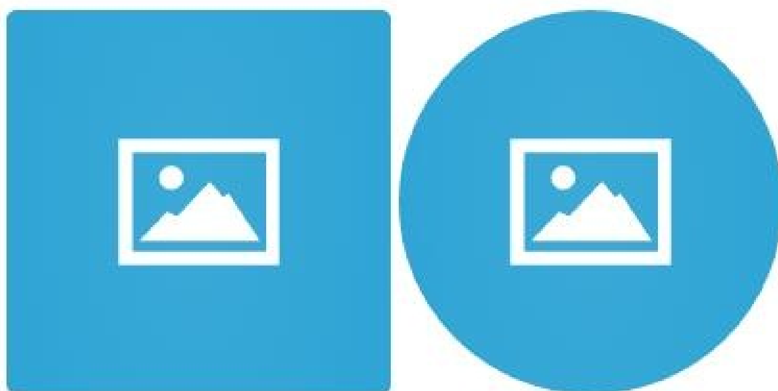
移除块元素内的 padding ，添加以下类中的一个就行了。

Class	描述
`.uk-padding-remove`	移除所有padding.
`.uk-padding-top-remove`	移除顶部padding.
`.uk-padding-bottom-remove`	移除底部padding
`.uk-padding-vertical-remove`	移除顶部和底部padding.

Border 半径

要为元素添加圆角，添加 `.uk-border-rounded` 即可。要使用圆形，添加 `.uk-border-circle` 即可。

Example



```
<img class="uk-border-rounded" src="" alt="">
<img class="uk-border-circle" src="" alt="">
```

标题变大

为了增大平板电脑和台式机下标题字体的大小，只需添加 `.uk-heading-large` 类。

示例

Heading

Code

```
<h1 class="uk-heading-large">...<h1>
```

哑色链接

如果要去掉链接的默认颜色，只需添加 `.uk-link-muted` 类到锚元素或它的父元素。

示例



Code

```
<a class="uk-link-muted">...<a>

<h1 class="uk-link-muted"><a>...<a><h1>

<ul class="uk-link-muted">
    <li><a>...<a></li>
    <li><a>...<a></li>
    <li><a>...<a></li>
</ul>
```

可滚动的预格式化文本

添加 `.uk-scrollable-text` 类设置一个最大高度，并提供一个垂直滚动条。这对预格式化的文本是非常有用的，它可以让你的代码块节省更多的空间。

示例

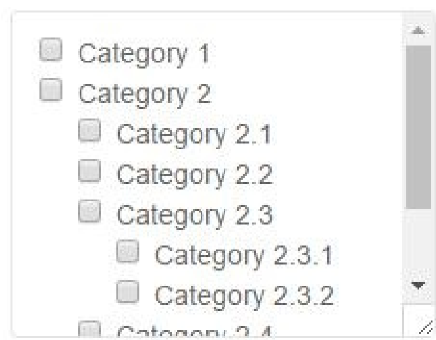
```
<!-- This is sample code to demonstrate preformatted text with a scrollbar. -->

<div class="uk-grid">
    <div class="uk-width-medium-1-2">
        <div class="uk-panel uk-panel-box uk-text-left">...</div>
    </div>
    <div class="uk-width-medium-1-2">
        <div class="uk-panel uk-panel-box uk-text-right">...</div>
    </div>
</div>
<div class="uk-grid">
    <div class="uk-width-medium-1-2">
        <div class="uk-panel uk-panel-box uk-text-center">...</div>
    </div>
    <div class="uk-width-medium-1-2">
        <div class="uk-panel uk-panel-box uk-text-justify">...</div>
    </div>
```

可滚动的盒子

添加 `.uk-scrollable-box` 类创建一个具有最大高度及垂直滚动条的看起来像面板的盒子。它可以包含任何类型的内容。

示例



Code

```
<div class="uk-scrollable-box">
    <ul class="uk-list">
        <li><label><input type="checkbox">...</label></li>
        <li><label><input type="checkbox">...</label></li>
    </ul>
</div>
```

溢出容器/Overflow container

当容器内部的元素宽度超过了容器本身，只需要为容器的 `<div>` 元素添加一个 `.uk-overflow-container` 类，就能为容器带来一个水平方向的滚动条。在响应式网页中处理表格时很有用，因为表格可能在某些断点会显得过于宽大。

示例

Table Heading	Table Heading	Table Heading	Table Heading	Table Heading	Table Headin
Table Data	Table Data	Table Data	Table Data	Table Data	Table Data
Table Data	Table Data	Table Data	Table Data	Table Data	Table Data
Table Data	Table Data	Table Data	Table Data	Table Data	Table Data
Table Footer	*Table Footer*	*Table Footer*	*Table Footer*	*Table Footer*	*Table Footer*

Code

```
<div class="uk-overflow-container">...</div>
```

显示效果

添加这些类中的一个改变元素的 display 属性。

Class	描述
`.uk-display-block`	强制将元素改变成块元素。
`.uk-display-inline`	强制将元素改变成内联元素。
`.uk-display-inline-block`	强制将元素改变成内联块元素。

可见性效果

Class	描述
`.uk-hidden`	在所有设备上隐藏该元素。
`.uk-hidden-touch`	在触控设备上隐藏
`.uk-hidden-notouch`	在非触控设备上隐藏
`.uk-invisible`	隐藏该元素，但是不在流量上删除该元素。
`.uk-visible-hover`	悬停时通过 `display: block` 来显示隐藏的内容。将这个类添加到父元素中。
`.uk-visible-hover-inline`	悬停时通过 `display: inline-block` 来显示隐藏的内容。将这个类添加到父元素中。

Example

来，悬停我...

Markup

```
<div class="uk-visible-hover">
    <div class="uk-hidden">...</div>
</div>
```

响应式可见性

你可以在特定的视口宽度下对内容进行显示或隐藏。通过设置断点变量可以很容易的进行修改。由于不同设备的尺寸变得越来越模糊，所以类的名称保持通用性而不提及任何具体的设备名称。

Class	Small (手机) 最大 767	Medium (平板) 768 到 959	Large (PC) 960 以上
`.uk-visible-small`	可见Visible	隐藏Hidden	隐藏Hidden
`.uk-visible-medium`	隐藏Hidden	可见Visible	隐藏Hidden
`.uk-visible-large`	隐藏Hidden	隐藏Hidden	可见Visible
`.uk-hidden-small`	隐藏Hidden	可见Visible	可见Visible
`.uk-hidden-medium`	可见Visible	隐藏Hidden	可见Visible
`.uk-hidden-large`	可见Visible	可见Visible	隐藏Hidden

Flex 布局

利用Flexbox的力量创建广泛的布局。

这个组件使用了Flexbox —— 一个比较新的概念，它拥有强大的布局效果。

用法

若要使用这个组件，只需要添加 `.uk-flex` 类到一个 `<div>` 元素。这样将会创建flex容器。默认地，所有flex条目都会左对齐，并被赋予一致的高度和宽度。

Example



Markup

```
<div class="uk-flex">
    <div>...</div>
</div>
```

行内的Flex

默认情况下，flex容器显示为块元素。为了仍然保持按照flexbox模型对其内容进行布局，并赋予其行内元素的行为，需要用到 `.uk-flex-inline` 类来替代 `uk-flex` .

修饰

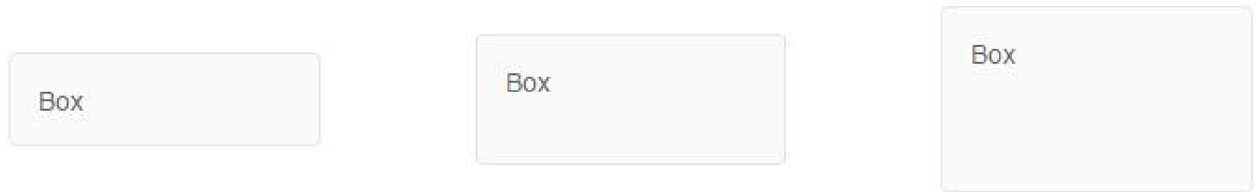
你可以添加多个不同的类来调整flex的行为。

对齐

这些类定义flex条目的水平或垂直对齐，并赋予它们彼此之间的间距。

Class	描述
`.uk-flex-center`	添加这个类，水平居中flex条目
`.uk-flex-right`	添加这个类，右对齐flex条目
`.uk-flex-top`	添加这个类，顶部对齐flex条目
`.uk-flex-middle`	添加这个类，垂直居中flex条目
`.uk-flex-bottom`	添加这个类，底部对齐flex条目
`.uk-flex-space-between`	添加这个类，使得条目均匀分布，第一个条目在主轴的开头，最后一个条目在主轴的末尾。
`.uk-flex-space-around`	添加这个类，使得条目均匀分布,使每个条目具有相同的左右空间。

Example



Markup

```
<div class="uk-flex uk-flex-middle uk-flex-space-between">...</div>
```

方向

这些类用于定义flex主轴的方向。默认地，flex条目按照水平从左到右的方向放置。

Class	描述
`.uk-flex-row-reverse`	添加这个类，使flex条目从右到左排列。
`.uk-flex-column`	添加这个类，使flex条目垂直排列成一列。
`.uk-flex-column-reverse`	添加这个类，使flex条目从下到上排列。

Example



Markup

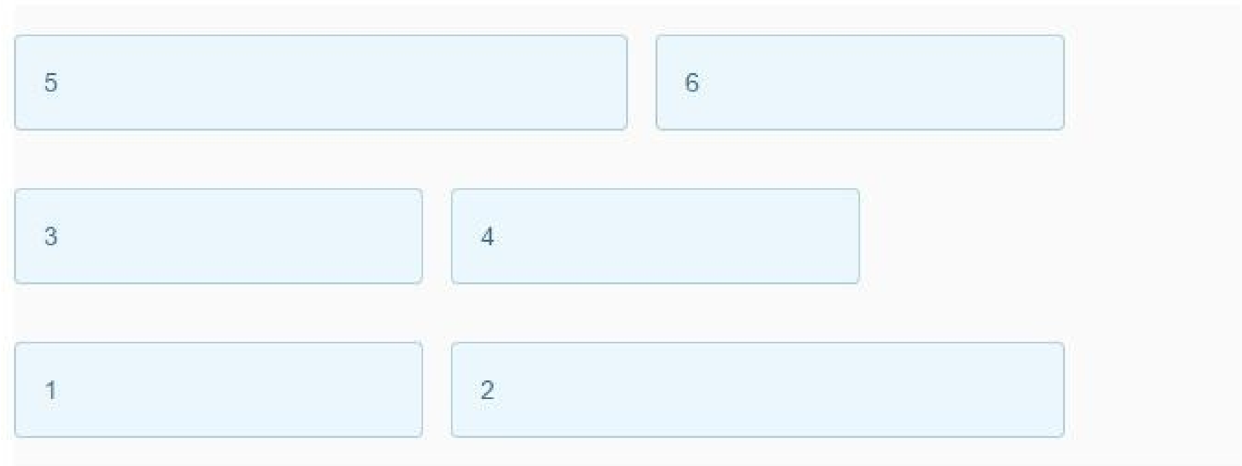
```
<div class="uk-flex uk-flex-column uk-flex-column-reverse">...</div
```

换行

默认情况下，flex条目将它们自身拟合到一行中。添加 `.uk-flex-wrap` 类，使条目不再匹配视口时切换到另一行。要改变条目的方向，使它们从右到左排列，添加 `.uk-flex-wrap-reverse` 类即可。下面这些类用来修饰换行的flex条目的对齐属性。强制将 flex条目放入一行，添加 `.uk-flex-nowrap` 类即可。

Class	描述
`.uk-flex-wrap-top`	添加这个类，使多行flex条目对齐到顶部。
`.uk-flex-wrap-middle`	添加这个类，使多行flex条目垂直居中。
`.uk-flex-wrap-bottom`	添加这个类，使多行flex条目对齐到底部。
`.uk-flex-wrap-space-between`	添加这个类，使条目的行均匀分布，第一行在容器顶部，最后一行在容器底部。
`.uk-flex-wrap-space-around`	添加这个类，是条目的行均匀分布，每一行都有一样的空间。

Example



Markup

```
<div class="uk-flex uk-flex-wrap uk-flex-wrap-reverse uk-flex-wrap-
```

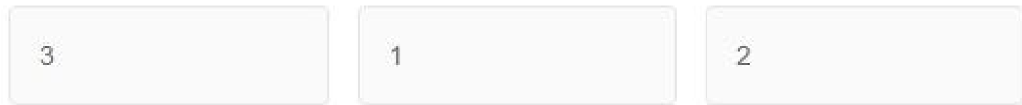
条目排序

默认地，flex条目根据源码的顺序排列。要将某个元素作为第一个或者最后一个进行显示，只需要添加下列类名中的一个。

Class	描述
`.uk-flex-order-first`	将此条目显示为第一个
`.uk-flex-order-last`	将此条目显示为最后一个
`.uk-flex-order-first-small`	

`.uk-flex-order-last-small` | 作用于视口宽度 *480px* 以上设备。 || `.uk-flex-order-first-medium` `.uk-flex-order-last-medium` | 作用于视口宽度 *768px* 以上设备。 || `.uk-flex-order-first-large` `.uk-flex-order-last-large` | 作用于视口宽度 *960px* 以上设备。 || `.uk-flex-order-first-xlarge` `.uk-flex-order-last-xlarge` | 作用于视口宽度 *1220px* 以上设备。 |

Example



Markup

```
<div class="uk-flex">
   <div class="uk-flex-order-first">...</div>
</div>
```

条目的规模

要确定一个flex条目需要占用多大的空间，为条目添加以下类中的一个即可。

Class	Description
`.uk-flex-item-none`	由内容决定其尺寸
`.uk-flex-item-auto`	按条目的内容分配空间
`.uk-flex-item-1`	空间分配完全基于Flex

Flex与网格

Flex组件可以与 网格 组合使用。

Example



Lorem ipsum dolor sit amet, consectetur adipisicing elit. Lorem ipsum dolor sit amet, consectetur adipisicing elit. Lorem ipsum dolor sit amet, consectetur adipisicing elit. Lorem ipsum dolor sit amet, consectetur adipisicing elit. Lorem ipsum dolor sit amet, consectetur adipisicing elit.

覆盖/Cover

扩展图片或视频至覆盖整个容器。

这个组件允许你使用图片、对象甚至iframe（images, objects or even iframes）来创建全屏效果。无论是什么元素，它都将垂直居中、水平居中并且不会失去原有的比例即实现覆盖它的容器。你还可以在图片或者视频上面放入附加内容，比如文字或图片等。

用法

这个覆盖组件具有不同的用法，取决于你究竟是使用的背景图片、对象或者iframe。最简单的方式就是为带有背景图片的 `<div>` 元素添加 `.uk-cover-background` 类。

Example



Markup

```
<div class="uk-cover-background">...</div>
```

视频

创建一个覆盖它的父容器的视频，添加 `.uk-cover-object` 类到视频。然后用一个容器元素包裹视频并为该容器添加 `.uk-cover` 类来裁剪内容。

Example

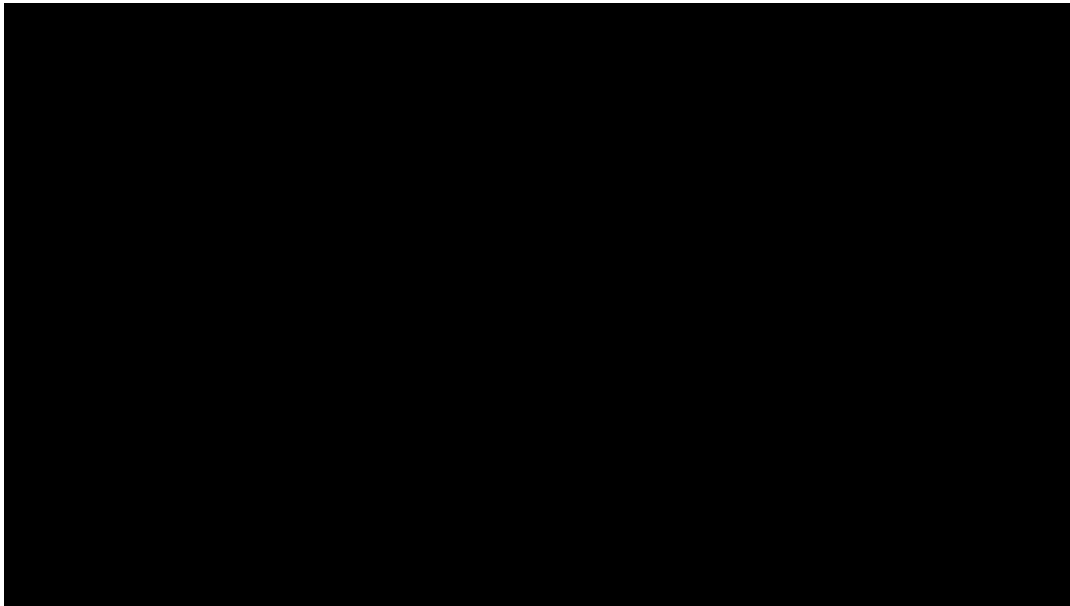
Markup

```
<div class="uk-cover">
    <video class="uk-cover-object" width="" height="">
        <source src="" type="">
    </video>
</div>
```

Iframe

要将覆盖组件应用到 iframe ，你只需要为 iframe 添加 `data-uk-cover` 属性。然后，再添加 `.uk-cover` 类到包含iframe的容器来裁剪内容。

Example



Markup

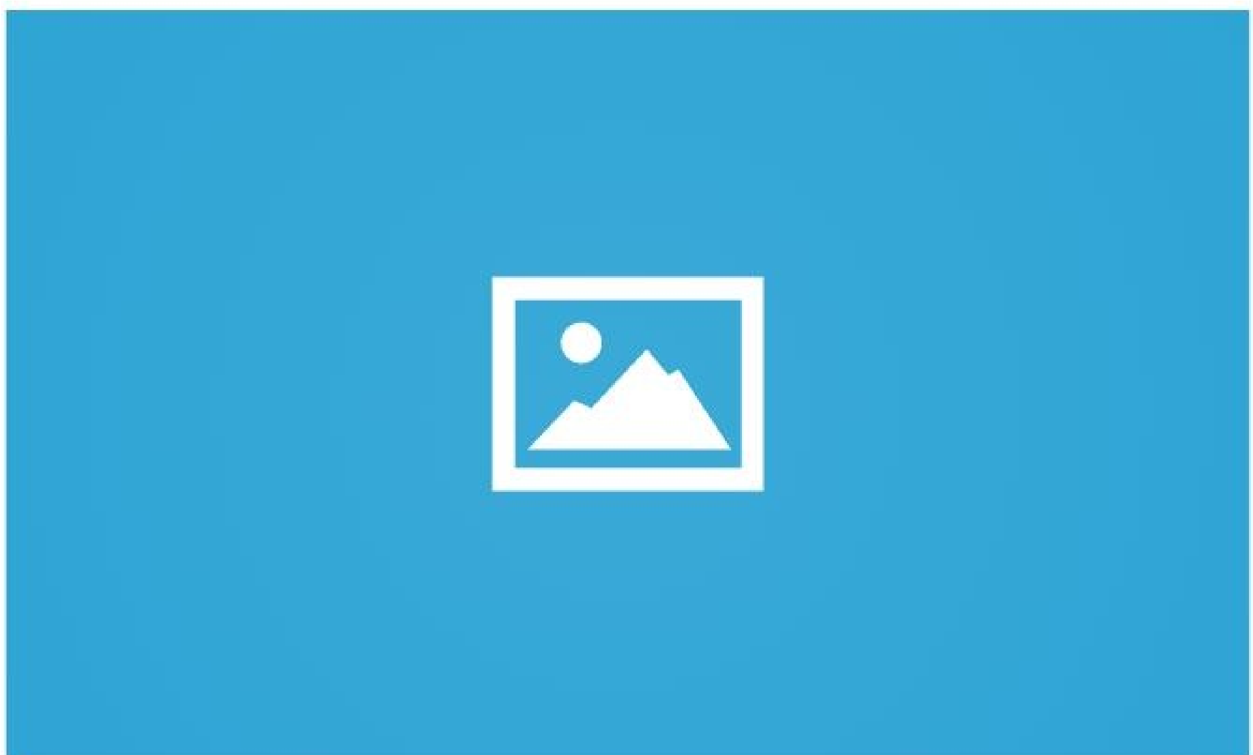
```
<div class="uk-cover">
    <iframe data-uk-cover src="" width="" height="" frameborder="0'
</div>
```

响应式

为覆盖图片添加响应式行为，你需要添加 `.uk-invisible` 类到 `<img>` 元素，并将它放在覆盖元素内部。这样的话，它就能适应图片的响应式行为了。

注意 添加 效果组件 中的 `.uk-height-viewport` 类，会扩展父容器器的高度填满整个视口。

Example



Markup

```
<div class="uk-cover-background">
    <img class="uk-invisible" src="" width="" height="" alt="">
</div>
```

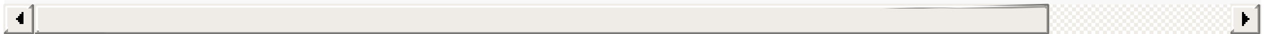
视频

为视频添加响应式行为，你同样需要为覆盖容器添加 `.uk-position-relative` 类，并将 `.uk-position-absolute` 类添加到覆盖对象上。对于irame也是这样操作。

Example

Markup

```
<div class="uk-cover uk-position-relative">
    <img class="uk-invisible" src="" width="" height="" alt="">
    <video class="uk-cover-object uk-position-absolute" width="" he
        <source src="" type="">
    </video>
</div>
```



内容的定位/Position content

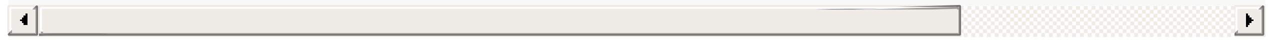
你还能在覆盖元素上面绝对定位内容。要实现这个效果，只需添加 效果组件 中的 `.uk-position-cover` 类到图片或视频后面的容器元素。如果想要实现垂直居中并且水平居中，那就使用 Flex 组件 吧。

Example



Markup

```
<div class="uk-cover-background uk-position-relative">
    <img class="uk-invisible" src="" width="" height="" alt="">
    <div class="uk-position-cover uk-flex uk-flex-center uk-flex-mi
</div>
```



导航类组件

导航菜单

为列表导航菜单定义不同的风格样式。

用法

使用这个组件，添加 `.uk-nav` 类到一个 `<ul>` 元素中。使用 `<a>` 元素作为列表中的菜单项。要对一个菜单项应用选中状态的效果，添加 `.uk-active` 类即可。

示例



注意 默认情况下，导航菜单没有任何样式。这就是为什么要添加一些具有样式的修饰类，这是很重要的，在例子中我们使用了 `.uk-nav-side` 类。

Code

```
<ul class="uk-nav">
    <li><a href="">...</a></li>
    <li class="uk-active"><a href="">...</a></li>
</ul>
```

嵌套导航

你可以很容易向菜单导航添加任意数量的 `<ul>` 元素。

Class	描述
`.uk-nav-sub`	在第一个嵌套的 `<ul>` 中添加这个类，使其拥有最佳的间距。
`.uk-parent`	添加这个类来表示一个父菜单项。
`.uk-nav-parent-icon`	给导航菜单添加这个类，可以使父菜单项增加图标。

示例



Code

```
<ul class="uk-nav uk-nav-parent-icon">
    <li class="uk-parent"><a href="">...</a>
        <ul class="uk-nav-sub">
            <li><a href="">...</a>
                <ul>...</ul>
            </li>
        </ul>
    </li>
</ul>
```

手风琴

默认情况下，子菜单项总是可见的。若要应用手风琴效果，只需给主 `<ul>` 元素添加 `data-uk-nav` 属性。当点击一个父菜单项时，会关闭已经展开的菜单项，每次只允许展开一个嵌套的菜单。若要避免这种情况，只需附加 `{multiple:true}` 到 data 属性。

示例



Code

```
<ul class="uk-nav uk-nav-parent-icon" data-uk-nav>
    <li class="uk-parent">
        <a href="">...</a>
        <ul class="uk-nav-sub">
            <li><a href="">...</a></li>
            <li><a href="">...</a></li>
        </ul>
    </li>
</ul>
```

```
<ul class="uk-nav uk-nav-parent-icon" data-uk-nav="{multiple:true}'
    <li class="uk-parent">
        <a href="">...</a>
        <ul class="uk-nav-sub">
            <li><a href="">...</a></li>
            <li><a href="">...</a></li>
        </ul>
    </li>
</ul>
```

标题和分隔线

为菜单项添加下面这些类中的一个，创建菜单项中的标题或分隔线。

Class	描述
.uk-nav-header	添加这个类到一个 元素创建一个标题。
.uk-nav-divider	添加这个类到一个 元素，在单独的菜单项之间创建一个分割线。

注意 你还可以给菜单项添加副标题。只需在菜单项的 <a> 元素里面创建一个 <div> 元素。

示例



Code

```
<li class="uk-nav-header">...</li>
<li class="uk-nav-divider"></li>
```

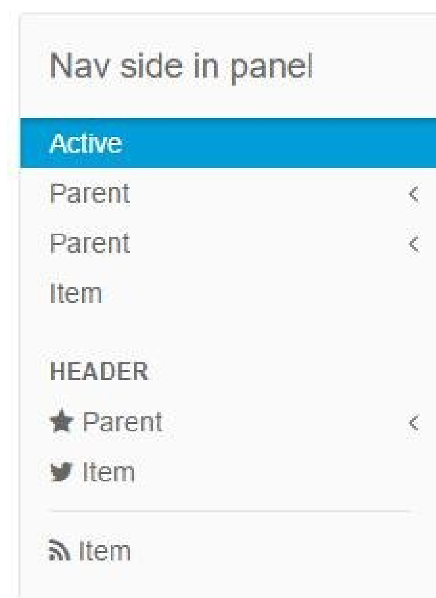
样式修饰

这里有一些修饰类，根据使用对象的上下文赋予导航菜单适当的样式。

侧边栏导航菜单

添加 `.uk-nav-side` 类，将导航菜单放置到侧边栏、面板或者网页中任意其他位置。

Example



Markup

```
<div class="uk-panel uk-panel-box">
    <h3 class="uk-panel-title">...</h3>
    <ul class="uk-nav uk-nav-side">...</ul>
</div>
```

下拉导航菜单

添加 `.uk-nav-dropdown` 类，将导航菜单放置到一个 下拉菜单组件 中的默认下拉菜单中。

示例



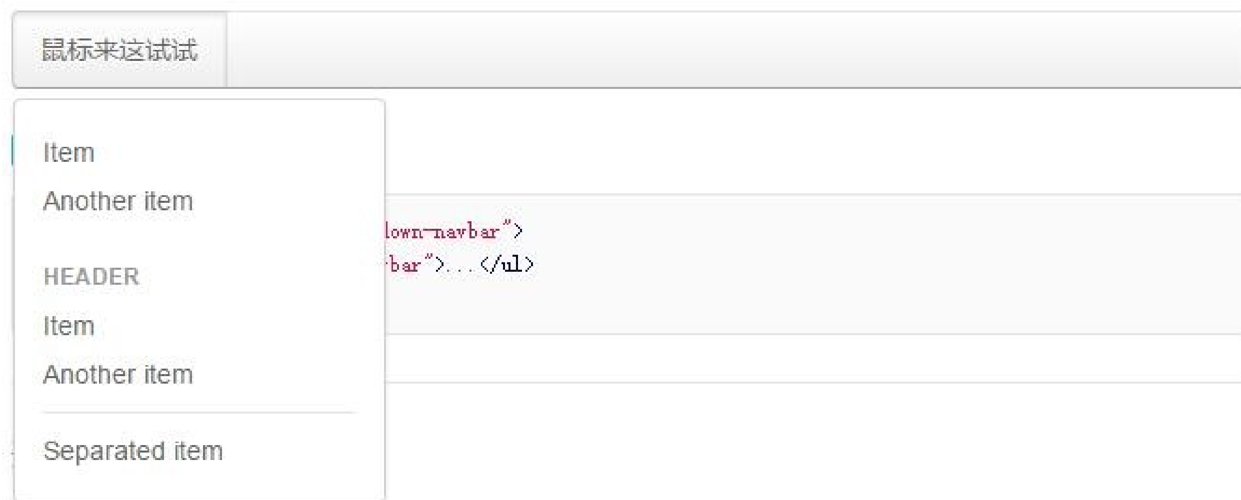
Code

```
<div class="uk-dropdown">
    <ul class="uk-nav uk-nav-dropdown">...</ul>
</div>
```

导航栏中的导航菜单

添加 `.uk-nav-navbar` 类，将导航菜单放置到一个 导航栏组件 中的导航栏下拉菜单中。

示例



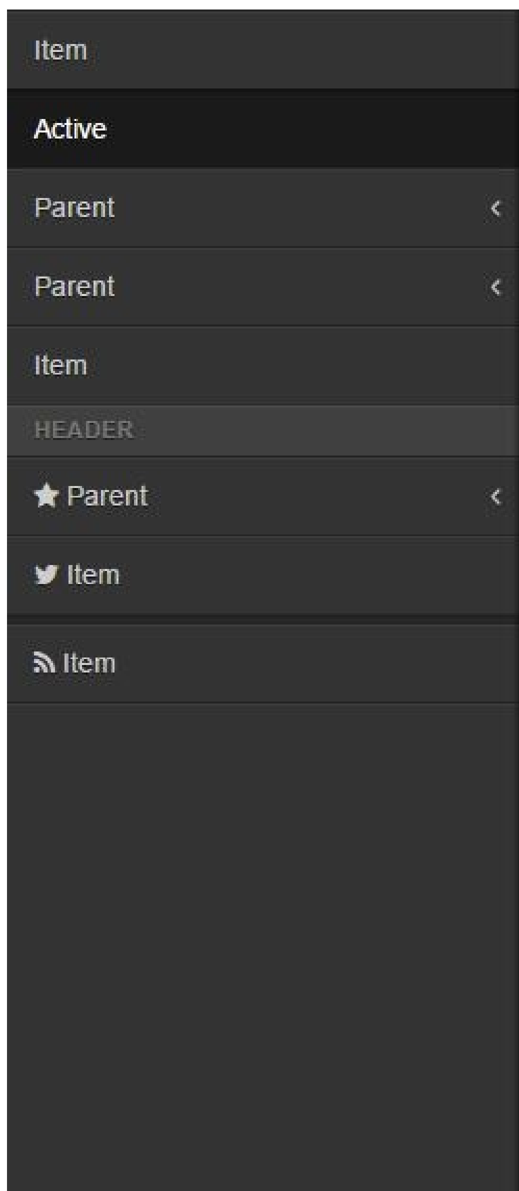
Code

```
<div class="uk-dropdown uk-dropdown-navbar">
    <ul class="uk-nav uk-nav-navbar">...</ul>
</div>
```

抽屉组件中的导航菜单

添加 `.uk-nav-offcanvas` 类，将导航菜单放置到 抽屉组件 里的侧边栏弹出式画布中。

示例



Code

```
<div class="uk-offcanvas-bar">
    <ul class="uk-nav uk-nav-offcanvas">...</ul>
</div>
```

导航栏

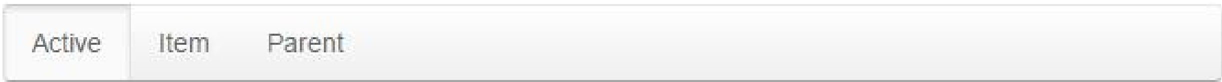
为导航栏定义不同风格的样式。

用法

导航栏组件包括导航栏本身及一个或多个导航菜单。

Class	描述
`.uk-navbar`	添加这个类到 <nav> 元素中，定义导航栏组件。
`.uk-navbar-nav`	添加这个类到 元素来创建导航。使用 <a> 元素作为列表中的菜单项。
`.uk-active`	添加这个类到一个列表项中，对其应用选中的状态。
`.uk-parent`	添加这个类来表示一个父级菜单项。

示例



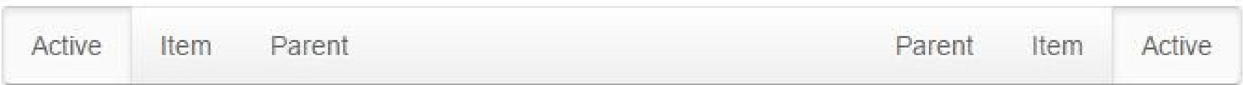
Code

```
<nav class="uk-navbar">
    <ul class="uk-navbar-nav">
        <li class="uk-active"><a href="">...</a></li>
        <li><a href="">...</a></li>
        <li class="uk-parent"><a href="">...</a></li>
    </ul>
</nav>
```

导航栏翻转

添加 `.uk-navbar-flip` 类到一个 <div> 元素中，将其进行分组并将导航项对齐到导航栏右侧。

示例



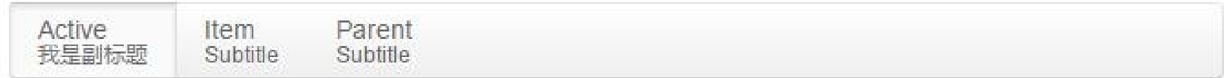
Code

```
<nav class="uk-navbar">
    <div class="uk-navbar-flip">
        <ul class="uk-navbar-nav">
            <li><a href="">...</a></li>
        </ul>
    </div>
</nav>
```

导航栏副标题

为列表项中的 <a> 元素添加 .uk-navbar-nav-subtitle 类来表示一个副标题。使用 <div> 元素作为副标题本身。

示例



Code

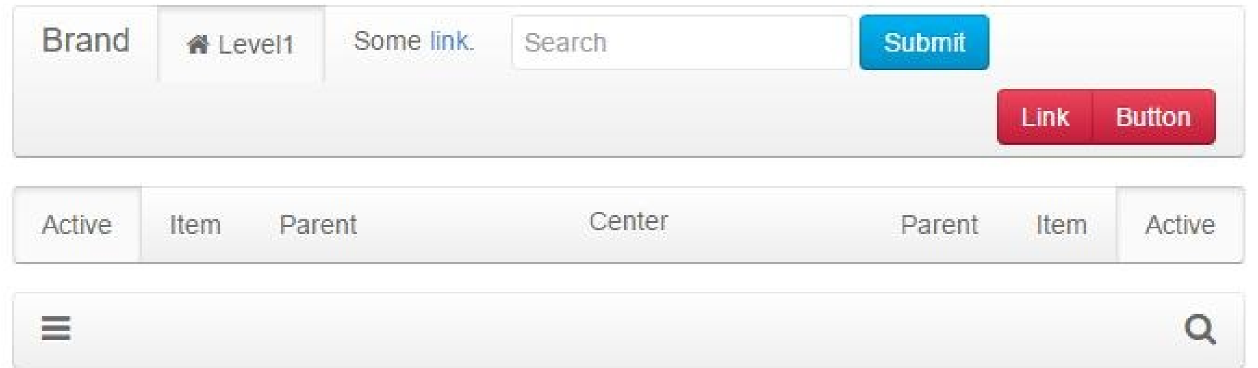
```
<nav class="uk-navbar">
    <ul class="uk-navbar-nav">
        <li><a href="" class="uk-navbar-nav-subtitle">
            ...
            <div>...</div>
        </a></li>
    </ul>
</nav>
```

内容

你还可以在导航栏中添加自定义的内容，如文本、图标、按钮、表单，甚至是一个拨动器。默认情况下，拨动器并没有附属 JavaScript。举例来说，可以为导航栏添加一个 抽屉组件 。

Class	描述
.uk-navbar-content	添加这个类到 <div> 元素中，使它作为一个自定义内容的容器。内容将被垂直居中。
.uk-navbar-brand	添加这个类到一个 <a> 或 <div> 元素中，来表示你的品牌。
.uk-navbar-center	添加这个类来居中你的导航栏内容或品牌。该元素必须为导航栏的最后一个子元素。
.uk-navbar-toggle	添加这个类到一个 <a> 或 <div> 元素中，创建一个作为拨动开关的图标。
.uk-navbar-toggle-alt	添加这个类来替换 拨动开关 默认的图标。

示例



Code

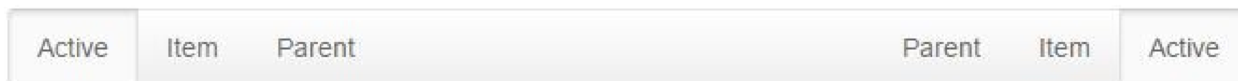
```
<nav class="uk-navbar">
    <a href="" class="uk-navbar-brand">...</a>
    <ul class="uk-navbar-nav">...</ul>
    <div class="uk-navbar-content">...</div>
    <div class="uk-navbar-content uk-navbar-center">...</div>
    <a href="" class="uk-navbar-toggle"></a>
</nav>
```

响应式可见性

要隐藏或显示导航栏中的某些项目，只需添加 效果组件 中的响应式可见性类。这是非常方便的，比如你想在移动设备中使用一个拨动器来触发导航。

示例

调整你的浏览器窗口大小可以看到它是怎么运行的。



Lorem ipsum dolor sit amet, consetetur sadipscing elitr.

Code

```
<nav class="uk-navbar">
    <ul class="uk-navbar-nav uk-hidden-small">...</ul>
    <a href="#my-id" class="uk-navbar-toggle uk-visible-small" data
</nav>

<div id="my-id" class="uk-offcanvas">
    ...
</div>
```

注意 在这个示例中，我们从 抽屉组件 中添加了一个侧边栏抽屉。

修饰

为了以不同的风格显示导航栏，只需添加修饰类。这些修饰类可以相互组合使用。

附着式的导航栏

添加 `.uk-navbar-attached` 类来优化附到视图顶部的导航栏样式。比如移除圆角。

示例



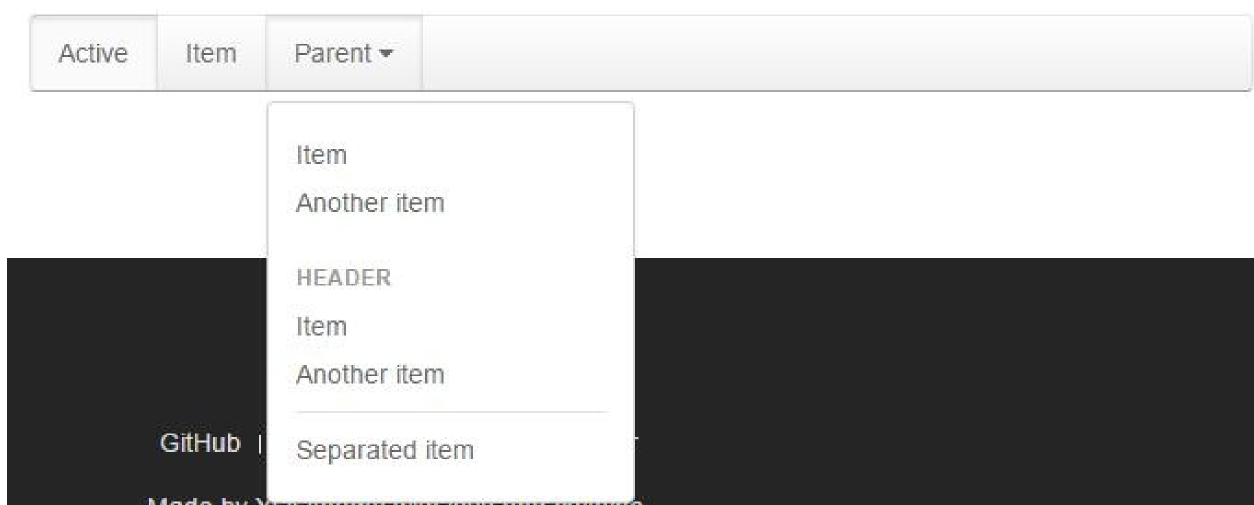
Code

```
<nav class="uk-navbar uk-navbar-attached">...</nav>
```

带有下拉菜单的导航栏

导航栏可以包含 下拉菜单组件 中的下拉菜单。只需添加 `.uk-dropdown-navbar` 修饰类到下拉菜单中，它可以完美的融入到导航栏的样式中。

示例



二级导航

为二级导航设置不同的样式风格

用法

使用以下类名来应用这个组件。对齐一个二级导航，比如水平居中，你可以使用 Flex 组件。

Class	描述
`.uk-subnav`	添加此类名到 `<ul>` 元素中，并在列表内嵌 `<a>` 元素。
`.uk-active`	为列表条目添加此类名，使其呈现选中状态。
`.uk-disabled`	为列表条目添加此类名，使其呈现禁用状态。

Example

Item Item Item Disabled

Markup

```
<ul class="uk-subnav">
    <li class="uk-active"><a href="">...</a></li>
    <li><a href="">...</a></li>
    <li class="uk-disabled"><a href="">...</a></li>
</ul>
```

修饰

二级导航的分隔线

添加 `.uk-subnav-line` 类名，用线条将菜单条目分隔开。

Example

Item | Item | Item | Disabled

Markup

```
<ul class="uk-subnav uk-subnav-line">
    <li>...</li>
</ul>
```

二级导航弹丸

添加 `.uk-subnav-pill` 类名，使选中状态的菜单条目拥有背景色。只需为菜单条目添加 `.uk-active` 类名，就能让它呈现选中的状态。

Example

Item Item Item Disabled

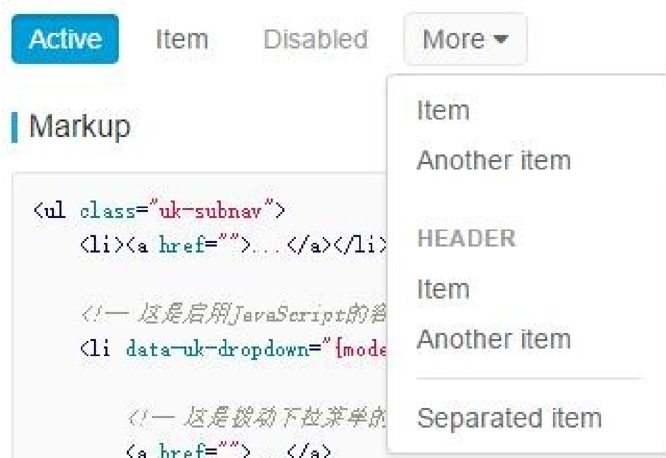
Markup

```
<ul class="uk-subnav uk-subnav-pill">
    <li class="uk-active">...</li>
</ul>
```

带下拉菜单的二级导航

这个例子展示了如何使用带有 下拉菜单 的二级导航。

Example



Markup

```
<ul class="uk-subnav">
    <li><a href="">...</a></li>

    <!-- 这是启用JavaScript的容器 -->
    <li data-uk-dropdown="{mode:'click'}">

        <!-- 这是拨动下拉菜单的导航项 -->
        <a href="">...</a>

        <!-- 这是下拉菜单 -->
        <div class="uk-dropdown uk-dropdown-small">
            <ul class="uk-nav uk-nav-dropdown">
                <li><a href="">...</a></li>
            </ul>
        </div>
    </li>
</ul>
```

面包屑/Breadcrumb

创建面包屑导航，显示用户在网站中的位置。

用法

面包屑组件由并排排列的链接和链接与链接直接的间隔组成。

类 Class	描述 Description
.uk-breadcrumb	在 元素中添加这个类定义面包屑组件。使用 <a> 元素作为列表中的面包屑项目。
.uk-active	在列表项中添加这个类，对其中一个应用一个选中的状态，并且使用 <span> 替代 <a> 元素即可。

使用一个 <span> 替代 <a> 元素，用来禁用某个面包屑项目。

示例

Home / Blog / Category / Post

Code

```
<ul class="uk-breadcrumb">
    <li><a href="">...</a></li>
    <li><a href="">...</a></li>
    <li><span>...</span></li>
    <li class="uk-active"><span>...</span></li>
</ul>
```

分页

轻松创建一个美观的分页导航来浏览网页。

用法

分页组件由类似按钮风格的链接并排排列组成。

类	概述
.uk-pagination	将这个类添加到一个 元素，定义分页导航组件。使用 <a> 元素作为分页导航列表中的项目。
.uk-active	将这个类添加到一个列表项，对其应用选中的状态，使用 <span> 替代 <a> 元素。
.uk-disabled	将这个类添加到一个列表项，对其应用禁用状态，使用 <span> 替代 <a> 元素。

要应用一个无样式的省略号，只需使用 <span> 来替代 <a> 元素。

示例



Code.

```
<ul class="uk-pagination">
    <li><a href="">...</a></li>
    <li class="uk-active"><span>...</span></li>
    <li class="uk-disabled"><span>...</span></li>
    <li><span>...</span></li>
</ul>
```

对齐修饰类

向页码添加 `.uk-pagination-left` 或 `.uk-pagination-right` 类使其向左或者向右对齐。

示例



Code.

```
<ul class="uk-pagination uk-pagination-left">
    <li>...</li>
</ul>
```

```
<ul class="uk-pagination uk-pagination-right">
    <li>...</li>
</ul>
```

上一页与下一页

创建一个简单上一页和下一页分页导航是非常容易的。只需添加 `.uk-pagination-previous` 或 `.uk-pagination-next` 类到一个 <li> 元素，其向左或者向右对齐。

示例



Code.

```
<li class="uk-pagination-previous"><a href="">...</a></li>
<li class="uk-pagination-next"><a href="">...</a></li>
```

带图标的分页导航

使用 图标组件 中的图标增强分页导航效果。添加一个 `.uk-icon-*` 类到分页导航链接里的 `<i>` 或 `<span>` 元素中。

示例



Code.

```
<li><a href=""><i class="uk-icon-angle-double-left"></i></a></li>
<li><a href=""><i class="uk-icon-angle-double-right"></i></a></li>
```

JavaScript

你可以应用附加组件中的分页组件来自动计算页码，例如在由Ajax支持的列表视图中触发一个事件，动态地加载新的列表项。

选项卡

创建拥有不同样式的选项卡导航。

用法

选项卡组件由一些并列的可点击选项卡标签组成。

Class类	描述
.uk-tab	添加这个类到一个 元素中定义一个选项卡组件。在列表中使用 <a> 元素作为选项卡标签。
.uk-active	添加这个类到选项卡标签，赋予选中状态。
.uk-disabled	添加这个类到选项卡标签，赋予禁用状态。

data-uk-tab 属性为两个目的提供支持。其一，它使得响应式行为成为可能。如果父容器太小而不能容纳所有的选项卡标签，它们将会合并到一个下拉菜单中，并由一个独立的默认选项卡作为拨动器。这里就需要用到 下拉菜单组件 了。

第二，它的功能耦合了 切换器组件,这是使用选项卡导航切换不同内容所必须的。

示例



Code

```
<ul class="uk-tab" data-uk-tab>
    <li class="uk-active"><a href="">...</a></li>
    <li><a href="">...</a></li>
    <li><a href="">...</a></li>
    <li class="uk-disabled"><a href="">...</a></li>
</ul>
```

水平方向的修饰

添加下列类中的一个，用来改变选项卡的外观。这些修饰类可以组合使用。

选项卡的对齐

Class类	描述
`.uk-tab-flip`	添加这个类，使选项卡右对齐，并翻转排列顺序。
`.uk-tab-bottom`	添加这个内到 `<ul>` ，将选项卡标签放在底部。这个类也能和其他类组合使用。

示例



Code

```
<ul class="uk-tab uk-tab-flip" data-uk-tab>...</ul>
<ul class="uk-tab uk-tab-bottom" data-uk-tab>...</ul>
```

调整选项卡标签

添加 `.uk-tab-grid` 类和一个 网格组件 中的 `.uk-width-*` 类，把选项卡标签放入网格中，使它的宽度与父元素宽度一致。

示例



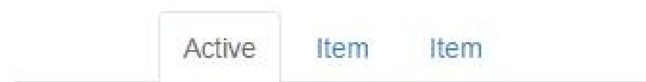
Code

```
<ul class="uk-tab uk-tab-grid uk-tab-bottom" data-uk-tab>
    <li class="uk-width-1-3"><a href="">...</a></li>
</ul>
```

选项卡标签居中

添加 `.uk-tab-center` 类到包含选项卡的 `<div>` 元素中，使选项卡居中。

示例



Code

```
<div class="uk-tab-center">
    <ul class="uk-tab" data-uk-tab>...</ul>
</div>
```

垂直方向的修饰

添加 `.uk-tab-left` 或 `.uk-tab-right` 类，将选项卡放置在左边或者右边。

示例



Code

```
<ul class="uk-tab uk-tab-left uk-width-medium-1-2">
    <li>...</li>
</ul>
```

```
<ul class="uk-tab uk-tab-right uk-width-medium-1-2">
    <li>...</li>
</ul>
```

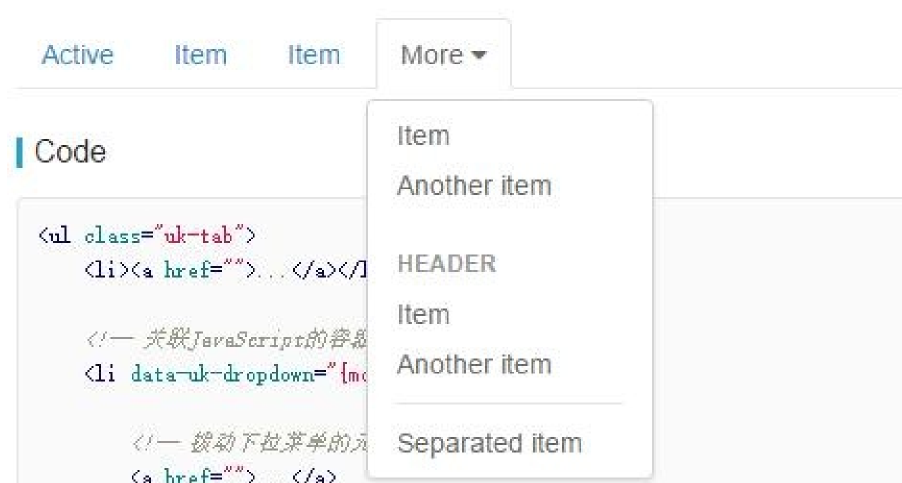
响应式行为

在狭窄的视口中，比如手机，垂直的选项卡会变成水平。添加 `data-uk-tab` 属性将为水平排列的选项卡赋予相同的响应行为。

含有下拉菜单的选项卡

这个例子是关于怎样使用带有 下拉菜单组件的选项卡。

示例



Code

```
<ul class="uk-tab">
    <li><a href="">...</a></li>

    <!-- 关联JavaScript的容器 -->
    <li data-uk-dropdown="{mode:'click'}">

        <!-- 拨动下拉菜单的元素 -->
        <a href="">...</a>

        <!-- 下拉菜单 -->
        <div class="uk-dropdown uk-dropdown-small">
            <ul class="uk-nav uk-nav-dropdown">
                <li><a href="">...</a></li>
            </ul>
        </div>
    </li>
</ul>
```

事件

你可以为以下事件绑定回调，以实现自定义功能：

名称	参数	描述
`change.uk.tab`	event, active item	On tab item change 选项卡切换时

缩略图导航/Thumbnav

创建可以使用 Flex 进行对齐的弹性缩略图导航。

用法

要使用这个组件的话，就这样做，添加 `.uk-thumbnav` 类到一个 `<ul>` 元素，然后将缩略图嵌套在列表条目内部的 `<a>` 元素中。如果缩略图导航比它的容器宽，并且使用了排水沟（gutter）时，添加 `.uk-active` 类创建active状态。

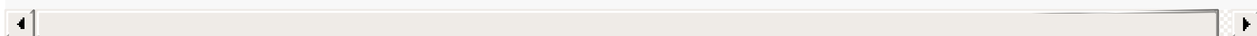
要对齐子导航的话，比如使它水平居中，你可以使用 Flex。

Example



Markup

```
<ul class="uk-thumbnav">
    <li class="uk-active"><a href=""><img src="" alt=""></a></li>
    <li><a href=""><img src="" alt=""></a></li>
</ul>
```



缩略图导航与网格

你可以使用 网格组件 中的 `.uk-grid-width-*` 类来实现在同一行中均匀地容纳缩略图导航条目。

Example



Markup

```
<ul class="uk-thumbnav uk-grid-width-1-5">
    <li>...</li>
</ul>
```

页面元素

列表

轻松创建具有不同风格的漂亮列表.

用法

要应用这个组件，添加 `.uk-list` 类到一个无序或有序列表即可。这时列表将不显示任何间距及列表样式.

示例

List item 1
List item 2
List item 3

code

```
<ul class="uk-list">
    <li>...</li>
    <li>...</li>
    <li>...</li>
</ul>
```

修饰

要为列表显示不同的样式，只需为 `.uk-list` 添加修饰类。列表组件的修饰类 不能 相互组合使用.

列表线

添加 `.uk-list-line` 类线分割列表项.

示例

List item 1
List item 2
List item 3

code

```
<ul class="uk-list uk-list-line">...</ul>
```

列表条纹

添加 `.uk-list-striped` 类列表条纹.

示例

- List item 1
- List item 2
- List item 3

code

```
<ul class="uk-list uk-list-striped">...</ul>
```

列表间距

添加 `.uk-list-space` 类增加列之间的间距.

Example

- 这个修饰类对于带有多行文本的列表项那是相当地有用
- Lorem ipsum dolor sit amet, consectetur adipisicing elit.
- Ut enim ad minim veniam, quis nostrud exercitation ullamco.

code

```
<ul class="uk-list uk-list-space">...</ul>
```

描述列表

轻松创建具有不同风格的漂亮的描述列表。

用法

创建描述列表并不需要像其他组件一样必须用到某个类，但是 UIkit 中提供了一些修饰类来定义不同风格样式的描述列表。描述列表组件的修饰类 不能 相互组合使用。

水平的描述列表

添加 `.uk-description-list-horizontal` 类使术语与描述并排显示。

示例

描述列表

描述列表用于定义术语和相应的描述。

描述列表 描述列表用于定义术语和相应的描述。
Lorem ipsum Dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.
被截断的一个比较长的... Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.

Code

```
<dl class="uk-description-list-horizontal">
    <dt>...</dt>
    <dd>...</dd>
</dl>
```

响应式行为

较窄的视口中，比如手机，描述列表将变回默认的堆叠样式。

描述列表线

添加 `.uk-description-list-line` 类用线条分隔描述列表项之间的间距。

示例

描述列表
描述列表用于定义术语和相应的描述。

Lorem ipsum
Dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Ut enim
Ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Code

```
<dl class="uk-description-list-line">...</dl>
```

表格

轻松创建具有不同风格的漂亮的表格。

用法

要应用这个组件，添加 `.uk-table` 类到一个 table 元素。表格的各行将被线条分隔开。

示例

表格说明/Table caption

表格标题	表格标题	表格标题
表格数据	Table Data	Table Data
Table Data	Table Data	Table Data
Table Data	Table Data	Table Data
表格脚注	*表格脚注*	*表格脚注*

Code

```
<table class="uk-table">
    <caption>...</caption>
    <thead>
        <tr>
            <th>...</th>
        </tr>
    </thead>
    <tfoot>
        <tr>
            <td>...</td>
        </tr>
    </tfoot>
    <tbody>
        <tr>
            <td>...</td>
        </tr>
    </tbody>
</table>
```

列

要修改列的宽度或内容，你可以使用其他组件。下面是一些实用的例子：

样式	描述
列宽	添加一个 网格组件 中的 `.uk-width-*` 类。
文本对齐	添加 文本组件 中的 `.uk-text-left` ，`.uk-text-right` 或 `.uk-text-center` 类。
文本样式	翻阅一下 文本组件。例如添加一个 `.uk-text-bold` 类。

Code

```
<td class="uk-width-1-10 uk-text-right">...</td>
<td class="uk-width-9-10 uk-text-bold">...</td>
```

垂直修饰

垂直居中表格内容，只需添加 `.uk-table-middle` 类到 `<tr>` 或 `<td>` 元素。

修饰

若要以不同的风格样式显示表格，只需添加一个修饰类到 `.uk-table` 类之后。

表格上的鼠标经过

添加 `.uk-table-hover` 类，赋予表格行鼠标经过状态。

示例

Table Heading	Table Heading	Table Heading
Table Data	Table Data	Table Data
Table Data	Table Data	Table Data
Table Data	Table Data	Table Data

Code

```
<table class="uk-table uk-table-hover">...</table>
```

表格条纹

添加 `.uk-table-striped` 类，为表格应用斑马线条纹状态。

示例

Table Heading	Table Heading	Table Heading
Table Data	Table Data	Table Data
Table Data	Table Data	Table Data
Table Data	Table Data	Table Data

Code

```
<table class="uk-table uk-table-striped">...</table>
```

压缩表格

添加 `.uk-table-condensed` 类使表格单元格更加紧凑。

示例

Table Heading	Table Heading	Table Heading
Table Data	Table Data	Table Data
Table Data	Table Data	Table Data
Table Data	Table Data	Table Data

Code

```
<table class="uk-table uk-table-condensed">...</table>
```

组合修饰类

表格组件的修饰类可以相互组合使用。

示例

Table Heading	Table Heading	Table Heading
Table Data	Table Data	Table Data
Table Data	Table Data	Table Data
Table Data	Table Data	Table Data

Code

```
<table class="uk-table uk-table-hover uk-table-striped uk-table-co
```

响应式表格

如果你的表格恰好比它的容器元素宽了一些或者最终在某个特定的视口宽度中显得太大，只需要用带有 `.uk-overflow-container` 类的 `<div>` 元素将其包裹。当内部的元素比这个容器本身宽的时候，将会创建一个水平的滚动条。

示例

Table Heading	Table Heading	Table Heading	Table Heading	Table Heading	Table Headin
Table Data	Table Data	Table Data	Table Data	Table Data	Table Data
Table Data	Table Data	Table Data	Table Data	Table Data	Table Data
Table Data	Table Data	Table Data	Table Data	Table Data	Table Data
Table Footer	*Table Footer*	*Table Footer*	*Table Footer*	*Table Footer*	*Table Footer*

Code

```
<div class="uk-overflow-container">
    <table>...</table>
</div>
```

表单

轻松创建拥有不同样式与布局的漂亮表单。

用法

为了应用这个组件，需要添加 `.uk-form` 类到表单元素中。所有表单控件都被并排放置在下面这行。

示例

Text input | Password input | Option 01 ▾ | Button | ☐ Checkbox

注意 在这个例子中，我们使用了一个按钮组件中的按钮。当表单元素在较小的视口中堆叠时，只需要添加 效果组件 中的 `data-uk-margin` 属性到它的父元素中，即可实现添加顶部margin。

Code

```
<form class="uk-form">

    <fieldset data-uk-margin>
        <legend>...</legend>
        <input type="text" placeholder="">
        <input type="password" placeholder="">
        <select>
            <option>...</option>
            <option>...</option>
        </select>
        <button class="uk-button">...</button>
        <label><input type="checkbox"> ...</label>
    </fieldset>

</form>
```

行

为一个 `<div>` 元素添加 `.uk-form-row` 用来放置表单控件。

示例

Legend

Text input

Password input

Option 01 ▼

Button

☐ Checkbox

Code

```
<form class="uk-form">
    <fieldset>
        <legend>...</legend>
        <div class="uk-form-row">...</div>
        <div class="uk-form-row">...</div>
    </fieldset>
</form>
```

控件的状态

通过表单控件上反馈的状态，为用户提供基础信息。

禁用

添加 `disabled` 属性到表单控件中，它的颜色会变淡，并禁止操作。

示例

Text input Option 01 ▼

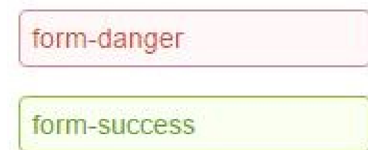
Code

```
<input type="text" placeholder="" disabled>
```

有效性状态

添加 `.uk-form-danger` 或 `.uk-form-success` 类到表单控件中，用于提示用户某个值是否通过有效性验证。

示例



Code

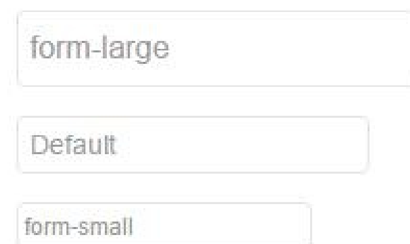
```
<input type="text" placeholder="" class="uk-form-danger">
<input type="text" placeholder="" class="uk-form-success">
```

控件的修饰

尺寸的调整

添加 `.uk-form-large` 或 `.uk-form-small` 类到 `<input>` , `<select>` 或 `<textarea>` 元素中使之变大或变小。

示例



Code

```
<input type="text" placeholder="" class="uk-form-large">
<input type="text" placeholder="" class="uk-form-small">
```

宽度的调整

添加 `.uk-form-width-large` , `.uk-form-width-medium` , `.uk-form-width-small` 或 `.uk-form-width-mini` 类到一个 `<input>` , `<select>` 或 `<textarea>` 元素中，调整它的宽度。

示例

form-width-large

form-width-medium

form-width-small

form

Code

```
<input type="text" placeholder="" class="uk-form-width-large">
<input type="text" placeholder="" class="uk-form-width-medium">
<input type="text" placeholder="" class="uk-form-width-small">
<input type="text" placeholder="" class="uk-form-width-mini">
```

你还可以在表单控件里使用 网格组件 中的 `.uk-width-*` 类。这是非常有用的，如果你想让表单控件的宽度扩展填满它的父级容器的宽度。

示例

width-100

Code

```
<input type="text" placeholder="" class="uk-width-1-1">
```

白板表单

添加 `.uk-form-blank` 类使表单控件的样式极简化。

示例

form-blank

Code

```
<input type="text" placeholder="" class="uk-form-blank">
```

帮助文本

使用 `.uk-form-help-inline` 或 `.uk-form-help-block` 类，为表单控件添加行内的或者块级的帮助文本。

示例

Text input 这里使用 `.uk-form-help-inline` 类建立了左侧的间距。

Textarea

这里使用 `.uk-form-help-block` 类建立了与上文相关联的段落。

Code

```
<div class="uk-form-row">
    <input type="text" placeholder=""> <span class="uk-form-help-i
</div>

<div class="uk-form-row">
    <textarea cols="" rows="" placeholder="">...</textarea>
    <p class="uk-form-help-block">...</p>
</div>
```

布局的调整

这里有两个可用的类可以用于布局调整： `.uk-form-stacked` and `.uk-form-horizontal` 。它们都要求表单空间被放置在带有 `.uk-form-row` 类的容器中。 Label 必须添加 `.uk-form-label` 类，并把相应的空间放置在带有 `.uk-form-controls` 类的容器中。

Code

```
<form class="uk-form uk-form-stacked">
    <div class="uk-form-row">
        <label class="uk-form-label" for="">...</label>
        <div class="uk-form-controls">...</div>
    </div>
</form>
```

注意 布局调整类同样可以用于 `<fieldset>` 元素中。这意味着如果你使用了多个fieldset，你可以为每个 fieldset设置不同的布局。

表单叠放

添加 `.uk-form-stacked` 类使label标签显示在控件上方。

示例



水平放置表单

添加 `.uk-form-horizontal` 类使label标签和控件水平并排放置。

示例

Text input Text input

Password input Password input

Select field Option 01

Textarea Textarea text

Radio input Radio input
1 2 3

Checkbox input Checkbox input
1 2 3

Mixed controls Checkbox input 5 Number input Option 01 Select field

表单控件中的文本

如果你在表单控件中同时使用了文本与单选框或者文本与复选框，只需要添加 `.uk-form-controls-text` 类就能使文本适当地对齐。

Code

```
<div class="uk-form-controls uk-form-controls-text">...</div>
```

表单控件中的垂直间距

如果你在表单控件中创建了段落，添加 `.uk-form-controls-condensed` 类可以减小段落间的间隔。

示例

Mixed controls混合放置的控件

☐ Checkbox input 5 Number input Option 01 ▾ Select field

☐ Checkbox input 5 Number input Option 01 ▾ Select field

Code

```
<div class="uk-form-controls uk-form-controls-text">
    <p class="uk-form-controls-condensed">...</p>
    <p class="uk-form-controls-condensed">...</p>
</div>
```

表单与网格

这个例子展示了如何用 网格组件 空间表单。

示例

100

50 25 25

50 50

Code

```
<form class="uk-form">

    <div class="uk-grid">
        <div class="uk-width-1-2"><input type="text" placeholder="'
        <div class="uk-width-1-4"><input type="text" placeholder="'
        <div class="uk-width-1-4"><input type="text" placeholder="'
    </div>

</form>
```

表单与图标

这个例子展示了怎么为表单添加 图标 。

示例



Code

```
<div class="uk-form-icon">
    <i class="uk-icon-calendar"></i>
    <input type="text">
</div>
```

表单增强

表单可以用 表单增强组件 进行扩展，以定义单选框与复选框的样式。

常用组件

按钮

轻松创建拥有不同样式的漂亮按钮。

用法

要应用这个组件，在一个 `<a>` 或 `<button>` 元素中添加 `.uk-button` 类即可。现在你已经创建了一个按钮。在 `<button>` 元素中添加 `disabled` 属性可以禁用按钮。

示例



Code

```
<a class="uk-button" href="">...</a>
<button class="uk-button" type="button">...</button>
<button class="uk-button" type="button" disabled>...</button>
```

注意 在一个组（row）里显示多个按钮，你可以给它们添加一个顶部外边距（top margin），这样,在较小的视口中它们会堆叠显示。仅需把效果组件 中的 `data-uk-margin` 属性添加到它们的父元素中即可。

色彩修饰

这里有几个颜色修饰类可以使用。仅需添加下列类中的一个，即可得到不同的外观。

示例	Class类	描述
Primary	`.uk-button-primary`	着重强调在一系列按钮中的重要性
Success	`.uk-button-success`	表示成功或积极的行为
Danger	`.uk-button-danger`	表示危险或负面的行为
Link	`.uk-button-link`	保持按钮的行为的同时，使其看起来更像一个链接。

尺寸修饰

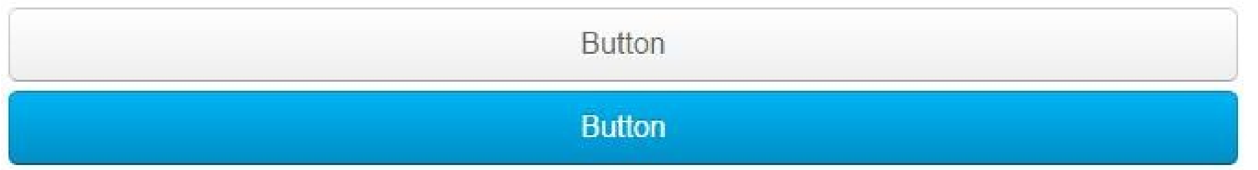
往一个按钮中添加 `.uk-button-mini` ，`.uk-button-small` 或 `.uk-button-large` 类使该按钮更小或更大。



满宽按钮

添加 网格组件 中的 `.uk-width-1-1` 类，该按钮将占满整个容器宽度。

示例



标签

```
<button class="uk-button uk-width-1-1 uk-margin-small-bottom">...</
<button class="uk-button uk-width-1-1">...</button>
```

按钮组

创建一个按钮组，在包裹着按钮们的 `<div>` 元素中添加 `.uk-button-group` 类即可。

示例



标签

```
<div class="uk-button-group">
    <a class="uk-button" href="">...</a>
    <button class="uk-button">...</button>
    <button class="uk-button">...</button>
</div>
```

JavaScript

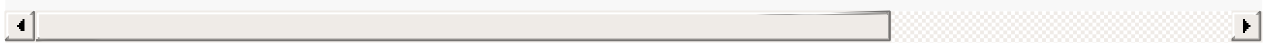
你可以通过JavaScript切换按钮状态。添加 `data-uk-button` 属性即可。

示例



标签

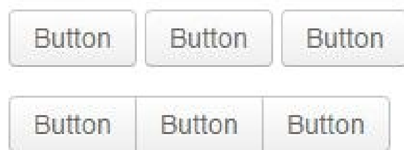
```
<button class="uk-button uk-button-primary" type="button" data-uk-b
```



复选按钮

像复选框一样切换一组按钮的状态，需要将这些按钮包裹在带有data属性 `data-uk-button-checkbox` 的 `<div>` 元素中。这属性也能用于按钮组。

示例



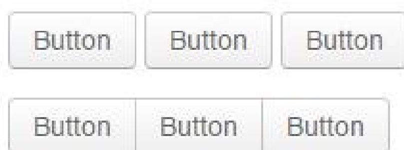
标签

```
<div data-uk-button-checkbox>
    <button class="uk-button">...</button>
    <button class="uk-button">...</button>
    <button class="uk-button">...</button>
</div>
```

单选按钮

想要按钮组像下面例子中一样只能按下一个？用带有 `data-uk-button-radio` 属性的 `<div>` 元素包裹一组按钮就行。这个也能用于按钮组。

示例



标签

```
<div data-uk-button-radio>
    <button class="uk-button">...</button>
    <button class="uk-button">...</button>
    <button class="uk-button">...</button>
</div>
```

带有下拉菜单的按钮

按钮可以用来触发 下拉菜单。只需要添加 `.uk-button-dropdown` 类名和 `data-uk-dropdown` 属性到包含按钮和下拉菜单的 `<div>` 元素中即可。

Example



Markup

```
<!-- This is the container enabling the JavaScript -->
<div class="uk-button-dropdown" data-uk-dropdown>

    <!-- This is the button toggling the dropdown -->
    <button class="uk-button">...</button>

    <!-- This is the dropdown -->
    <div class="uk-dropdown uk-dropdown-small">
        <ul class="uk-nav uk-nav-dropdown">
            <li><a href="">...</a></li>
            <li><a href="">...</a></li>
        </ul>
    </div>

</div>
```

带下拉菜单的按钮组

使用按钮组将按钮分成左边是标准行为的按钮和右边是带下拉菜单的按钮，仅需将他们用一个 `<div>` 元素包裹，然后在 `<div>` 中添加 `data-uk-dropdown="{mode:'click'}"` 属性即可。 当然，下拉菜单也可以用于按钮组中的一个按钮。

示例



Code

```
<div class="uk-button-group">

    <!-- 标准按钮 -->
    <button class="uk-button">...</button>

    <!-- 关联JavaScript的容器 -->
    <div data-uk-dropdown="{mode:'click'}">

        <!-- 触发下拉菜单的按钮 -->
        <a href="" class="uk-button">...</a>

        <!-- 下拉菜单 -->
        <div class="uk-dropdown uk-dropdown-small">
            <ul class="uk-nav uk-nav-dropdown">
                <li><a href="">...</a></li>
                <li><a href="">...</a></li>
            </ul>
        </div>

    </div>
</div>
```

图标

使用图标字体，在任何地方放置矢量图标。

这个组件使用了由Dave Gandy发起的 Font Awesome 图标字体。Font Awesome 为web相关行为提供了超过300个标志和符号。 由于允许你通过CSS简单地改变颜色、尺寸和其他样式，图标字体是很棒的选择。它们是可伸缩的矢量图形，这意味着它们在高分辨率设备上也能很好地显示。

用法

应用此组件，在任何 `<i>` 或 `<span>` 元素中添加 `.uk-icon-*` 类即可。瞧，你有了一个矢量图标，它能像文本一样继承了尺寸和色彩。

示例

使用 `.uk-icon-cog` 类

使用 `.uk-icon-user` 类

使用 `.uk-icon-home` 类

链接中的图标

带有图标的按钮

标签

```
<!-- 图标 -->
<i class="uk-icon-cog"></i>

<!-- 链接中的图标 -->
<a href=""><i class="uk-icon-cog"></i> ...</a>
```

更多示例

按钮组

这个示例展示了每个按钮都带有图标的按钮租，来自 按钮组件 。

示例



按钮下拉菜单

这个示例展示了一个被分割成左边是正常按钮右边是下拉菜单切换按钮的按钮，来自下拉菜单组件。

示例



块引用

这个示例展示基础组件 中个块引用，它使用了一个大引号图标，还使用了效果组件中的 `.uk-align-left` 类。

示例



尺寸

添加 `.uk-icon-small` ， `.uk-icon-medium` 或 `.uk-icon-large` 类使一个图标更大。

这是默认尺寸。This is the default size.

这个图标使用 `.uk-icon-small` 类

这个图标使用 `.uk-icon-medium` 类

这个图标使用 `.uk-icon-large` 类

旋转

添加 `.uk-icon-spin` 创建动画的旋转图标。Add the `.uk-icon-spin` class to create animated spinning icons.

用于加载内容的旋转图标 ...

用于刷新内容的图标

图标的 hover 效果

使用 `.uk-icon-hover` 修饰器将锚文本做成图标效果，它添加了hover效果并溢出了链接下划线。

Example



Markup

```
<a href="" class="uk-icon-hover uk-icon-github"></a>
```

图标按钮

使用修饰类 `.uk-icon-button` 创建一个图标按钮，可以完美地用于社交图标。

示例



标签

```
<a href="" class="uk-icon-button uk-icon-github"></a>
```

调整图标

为图标添加固定的宽度，并将其居中，只需添加 `.uk-icon-justify` 类名。在一个列表中使用不同的图标时是很有帮助的哦。

Example

- 调整过的图标
- 调整过的图标
- 调整过的图标
- 调整过的图标

Markup

```
<a href="" class="uk-icon-justify uk-icon-github"></a>
```

图标总览

这是所有一个所有可用的 `.uk-icon-*` 类的总览，由Font Awesome提供。

Web 应用程序图标

adjust	anchor	archive	area-chart
arrows	arrows-h	arrows-v	asterisk
at	automobile (alias)	balance-scale	ban
bank (alias)	bar-chart	bar-chart-o (alias)	barcode
bars	battery-empty (alias)	battery-0	battery-quarter (alia
battery-1	battery-half (alias)	battery-2	battery-three-quarte
battery-3	battery-full (alias)	battery-4	bed
beer	bell	bell-o	bell-slash
bell-slash-o	bicycle	binoculars	birthday-cake
bluetooth	bluetooth-b	bolt	bomb
book	bookmark	bookmark-o	briefcase
bug	building	building-o	bullhorn
bullseye	bus	cab (alias)	calculator
calendar	calendar-check-o	calendar-minus-o	calendar-o
calendar-plus-o	calendar-times-o	camera	camera-retro
car	caret-square-o-down	caret-square-o-left	caret-square-o-right
caret-square-o-up	cart-arrow-down	cart-plus	cc
certificate	check	check-circle	check-circle-o
check-square	check-square-o	child	circle
circle-o	circle-o-notch	circle-thin	clock-o
clone	close (alias)	cloud	cloud-download
cloud-upload	code	code-fork	coffee
cog	cogs	comment	comment-o
commenting	commenting-o	comments	comments-o
compass	copyright	creative-commons	credit-card
credit-card-alt	crop	crosshairs	cube
cubes	cutlery	dashboard (alias)	database
desktop	diamond	dot-circle-o	download
edit (alias)	ellipsis-h	ellipsis-v	envelope
envelope-o	envelope-square	eraser	exchange
exclamation	exclamation-circle	exclamation-triangle	external-link
external-link-square	eye	eye-slash	eyedropper
fax	female	fighter-jet	file-archive-o
file-audio-o	file-code-o	file-excel-o	file-image-o
file-movie-o (alias)	file-pdf-o	file-photo-o (alias)	file-picture-o (alias)
file-powerpoint-o	file-sound-o (alias)	file-video-o	file-word-o
file-zip-o (alias)	film	filter	fire
fire-extinguisher	flag	flag-checkered	flag-o
flash (alias)	flask	folder	folder-o
folder-open	folder-open-o	frown-o	futbol-o
gamepad	gavel	gear (alias)	gears (alias)
genderless (alias)	gift	glass	globe
graduation-cap	group (alias)	hand-lizard-o	hand-stop-o (alias)
hand-paper-o	hand-peace-o	hand-pointer-o	hand-grab-o (alias)
hand-rock-o	hand-scissors-o	hand-spock-o	hdd-o
hashtag	headphones	heart	heart-o
heartbeat	history	home	hotel (alias)
hourglass	hourglass-o	hourglass-1 (alias)	hourglass-start

hourglass-2 (alias)	hourglass-half	hourglass-3 (alias)	hourglass-end
i-cursor	image (alias)	inbox	industry
info	info-circle	institution (alias)	key
keyboard-o	language	laptop	leaf
legal (alias)	lemon-o	level-down	level-up
life-bouy (alias)	life-buoy (alias)	life-ring	life-saver (alias)
lightbulb-o	line-chart	location-arrow	lock
magic	magnet	mail-forward (alias)	mail-reply (alias)
mail-reply-all (alias)	male	map	map-marker
map-o	map-pin	map-signs	meh-o
microphone	microphone-slash	minus	minus-circle
minus-square	minus-square-o	mobile	mobile-phone (alias)
money	moon-o	mortar-board (alias)	motorcycle
mouse-pointer	music	navicon (alias)	newspaper-o
object-group	object-ungroup	paint-brush	paper-plane
paper-plane-o	paw	pencil	pencil-square
pencil-square-o	percent	phone	phone-square
photo (alias)	picture-o	pie-chart	plane
plug	plus	plus-circle	plus-square
plus-square-o	power-off	print	puzzle-piece
qrcode	question	question-circle	quote-left
quote-right	random	recycle	refresh
registered	remove (alias)	reorder (alias)	reply
reply-all	retweet	road	rocket
rss	rss-square	search	search-minus
search-plus	send (alias)	send-o (alias)	server
share	share-alt	share-alt-square	share-square
share-square-o	shield	ship	shopping-bag
shopping-basket	shopping-cart	sign-in	sign-out
signal	sitemap	sliders	smile-o
soccer-ball-o (alias)	sort	sort-alpha-asc	sort-alpha-desc
sort-amount-asc	sort-amount-desc	sort-asc	sort-desc
sort-down (alias)	sort-numeric-asc	sort-numeric-desc	sort-up (alias)
space-shuttle	spinner	spoon	square
square-o	star	star-half	star-half-empty (alias)
star-half-full (alias)	star-half-o	star-o	sticky-note
sticky-note-o	street-view	suitcase	sun-o
support (alias)	tablet	tachometer	tag
tags	tasks	taxi	television
terminal	thumb-tack	thumbs-down	thumbs-o-down
thumbs-o-up	thumbs-up	ticket	times
times-circle	times-circle-o	tint	toggle-down (alias)
toggle-left (alias)	toggle-off	toggle-on	toggle-right (alias)
toggle-up (alias)	trademark	trash	trash-o
tree	trophy	truck	tty
tv (alias)	umbrella	university	unlock
unlock-alt	unsorted (alias)	upload	usb
user	user-plus	user-secret	user-times
users	video-camera	volume-down	volume-off
volume-up	warning (alias)	wheelchair	wifi
wrench			

交通工具图标

ambulance	automobile (alias)	bicycle	bus
cab (alias)	car	fighter-jet	motorcycle
plane	rocket	ship	space-shuttle
subway	taxi	train	truck
wheelchair			

手势图标

hand-lizard-o	hand-stop-o (alias)	hand-paper-o	hand-peace-o
hand-pointer-o	hand-grab-o (alias)	hand-rock-o	hand-scissors-o
hand-spock-o	thumb-tack	thumbs-down	thumbs-o-down
thumbs-o-up	thumbs-up		

性别图标

genderless (alias)	mars	mars-double	mars-stroke
mars-stroke-h	mars-stroke-v	mercury	neuter
transgender	transgender-alt	venus	venus-double
venus-mars			

文件类型图标

file	file-archive-o	file-audio-o	file-code-o
file-excel-o	file-image-o	file-movie-o (alias)	file-o
file-pdf-o	file-photo-o (alias)	file-picture-o (alias)	file-powerpoint-o
file-sound-o (alias)	file-text	file-text-o	file-video-o
file-word-o	file-zip-o (alias)		

旋转图标

circle-o-notch	cog	gear (alias)	refresh
spinner			

表单控件图标

check-square	check-square-o	circle	circle-o
dot-circle-o	minus-square	minus-square-o	plus-square
plus-square-o	square	square-o	

支付工具图标

cc-amex	cc-diners-club	cc-discover	cc-jcb
cc-mastercard	cc-paypal	cc-stripe	cc-visa
credit-card	google-wallet	paypal	

货币



文本编辑器

align-center	align-justify	align-left	align-right
bold	chain (alias)	chain-broken	clipboard
columns	copy (alias)	cut (alias)	dedent (alias)
eraser	file	file-o	file-text
file-text-o	files-o	floppy-o	font
header	indent	italic	link
list	list-alt	list-ol	list-ul
outdent	paperclip	paragraph	paste (alias)
repeat	rotate-left (alias)	rotate-right (alias)	save (alias)
scissors	strikethrough	x_2 subscript	x^2 superscript
table	text-height	text-width	th
th-large	th-list	underline	undo
unlink (alias)			

方向

angle-double-down	angle-double-left	angle-double-right	angle-double-up
angle-down	angle-left	angle-right	angle-up
arrow-circle-down	arrow-circle-left	arrow-circle-o-down	arrow-circle-o-left
arrow-circle-o-right	arrow-circle-o-up	arrow-circle-right	arrow-circle-up
arrow-down	arrow-left	arrow-right	arrow-up
arrows	arrows-alt	arrows-h	arrows-v
caret-down	caret-left	caret-right	caret-square-o-down
caret-square-o-left	caret-square-o-right	caret-square-o-up	caret-up
chevron-circle-down	chevron-circle-left	chevron-circle-right	chevron-circle-up
chevron-down	chevron-left	chevron-right	chevron-up
hand-o-down	hand-o-left	hand-o-right	hand-o-up
long-arrow-down	long-arrow-left	long-arrow-right	long-arrow-up
toggle-down (alias)	toggle-left (alias)	toggle-right (alias)	toggle-up (alias)

视频播放器

arrows-alt	backward	compress	eject
expand	fast-backward	fast-forward	forward
pause	pause-circle	pause-circle-o	play
play-circle	play-circle-o	step-backward	step-forward
stop	stop-circle	stop-circle-o	youtube-play

品牌图标

- 500px
- adn
- amazon
- android
- angellist
- apple
- behance
- behance-square
- bitbucket
- bitbucket-square
- bitcoin (alias)
- black-tie
- bluetooth
- bluetooth-b
- btc
- buysellads
- cc-amex
- cc-diners-club
- cc-discover
- cc-jcb
- cc-mastercard
- cc-paypal
- cc-stripe
- cc-visa
- chrome
- codepen
- codiepie
- connectdevelop
- contao
- css3
- dashcube
- delicious
- deviantart
- digg
- dribbble
- dropbox
- drupal
- edge
- empire
- expeditedssl
- facebook
- facebook-f (alias)
- facebook-official
- facebook-square
- firefox
- flickr
- fonticons
- fort-awesome
- forumbee
- foursquare
- ge (alias)
- get-pocket
- gg
- gg-circle
- git
- git-square
- github
- github-alt
- github-square
- gittip (alias)
- google
- google-plus
- google-plus-square
- google-wallet
- gratipay
- hacker-news
- houzz
- html5
- instagram
- internet-explorer
- ioxhost
- joomla
- jsfiddle
- lastfm
- lastfm-square
- leanpub
- linkedin
- linkedin-square
- linux
- maxcdn
- meanpath
- medium-logo
- mixcloud
- modx
- odnoklassniki
- odnoklassniki-square
- opencart
- openid
- opera
- optin-monster
- pagelines
- paypal
- pied-piper
- pied-piper-alt
- pinterest
- pinterest-p
- pinterest-square
- product-hunt
- qq
- ra (alias)
- rebel
- reddit
- reddit-alien
- reddit-square
- renren
- safari
- scribd
- sellsy
- share-alt
- share-alt-square
- shirtsinbulk
- simplybuilt
- skyatlas
- skype
- slack
- slideshare
- soundcloud
- spotify
- stack-exchange
- stack-overflow
- steam
- steam-square
- stumbleupon
- stumbleupon-circle
- tencent-weibo
- trello
- tripadvisor
- tumblr
- tumblr-square
- twitch
- twitter
- twitter-square
- usb
- viacoin
- vimeo
- vimeo-square
- vine
- vk
- wechat (alias)
- weibo
- weixin
- whatsapp
- wikipedia-w
- windows
- wordpress
- xing
- xing-square
- y-combinator
- y-combinator-square
- yahoo
- yc (alias)
- yc-square (alias)
- yelp
- youtube
- youtube-play
- youtube-square

医疗类图标

- ambulance
- h-square
- hospital-o
- medkit
- plus-square
- stethoscope
- user-md
- wheelchair

关闭/Close

定义不同样式的，能与各种组件结合使用的关闭按钮。

用法

使用这个组件，需要添加 `.uk-close` 类到一个 `<a>` 或 `<button>` 元素中。

示例



Code

```
<a href="" class="uk-close"></a>
```

样式调整

添加 `.uk-close-alt` 类，为关闭按钮带来一个非同寻常的样式。

示例



Code

```
<a href="" class="uk-close uk-close-alt"></a>
```

提示框里的关闭按钮

这是一个关于如何将关闭按钮和提示框结合使用的例子。

示例

这是一个在提示框中使用关闭按钮的例子。 

这是一个在提示框中使用关闭按钮的例子。

模态对话框里的关闭按钮

这是一个在模态对话框中使用关闭组件的例子。

示例

Button

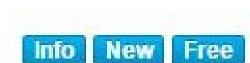
徽章/Badge

轻松创建好看的徽章来标示和强调你的内容。

用法

要创建一个徽章，只需添加 `.uk-badge` 类到一个 `<div>` 或 `<span>` 元素中。

示例



Code

```
<div class="uk-badge">...</div>
```

修饰

若需以不同的样式显示标签，只需为带有 `.uk-badge` 类的添加修饰类。这些修饰类可以相互结合使用。

通知

添加 `.uk-badge-notification` 类标示一个通知，通常用于显示数字。

示例



Code

```
<div class="uk-badge uk-badge-notification">...</div>
```

色彩修饰

添加 `.uk-badge-success` , `.uk-badge-warning` 或者 `.uk-badge-danger` 类为标签添加色彩。

示例



Code

```
<div class="uk-badge uk-badge-success">...</div>
<div class="uk-badge uk-badge-warning">...</div>
<div class="uk-badge uk-badge-danger">...</div>
```

提示框

为成功、警告和错误信息定义样式。

用法

要应用此组件，添加 `.uk-alert` 类至块元素中即可。

示例

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore.

标签

```
<div class="uk-alert">...</div>
```

关闭按钮

为了使用一个关闭按钮，在该提示框内部的 `<button>` 或 `<a>` 元素中添加 `.uk-alert-close` 类。要使关闭按钮的JavaScript生效，仅需往该提示框中添加 `data-uk-alert` 属性。虽然你也可以使用文本或者一幅图片，我们还是建议添加 关闭组件 中的 `.uk-close` 类来使这个按钮拥有一个更完美的样式。

示例

Lorem ipsum dolor sit amet, consectetur adipisicing elit.



标签

```
<div class="uk-alert" data-uk-alert>
    <a href="" class="uk-alert-close uk-close"></a>
    <p>...</p>
</div>
```

修饰

有几种不同的颜色修饰类可供选用。仅需添加以下类中的一个就能得到一个不同的外观。

示例

To indicate success or a positive message add the `.uk-alert-success` class. ×

To indicate a message containing a warning add the `.uk-alert-warning` class. ×

To indicate an important message add the `.uk-alert-danger` class. ×

标签

```
<div class="uk-alert uk-alert-success"> ... </div>
<div class="uk-alert uk-alert-warning">...</div>
<div class="uk-alert uk-alert-danger">...</div>
```

尺寸

添加 `.uk-alert-large` 类增大一个提示框的空间。如果你想使用更丰富的排版，这会非常有用。

示例

×

Today's headline

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

标签

```
<div class="uk-alert uk-alert-large">...</div>
```

缩略图/Thumbnail

创建拥有各种不同样式和尺寸的缩略图。

用法

要使应用个组件，只需要添加 `.uk-thumbnail` 类到一个 `<img>` ， `<a>` 或 `<figure>` 元素中。

示例



Code

```
<!-- This is an image as a thumbnail -->
<img class="uk-thumbnail" src="" alt="">

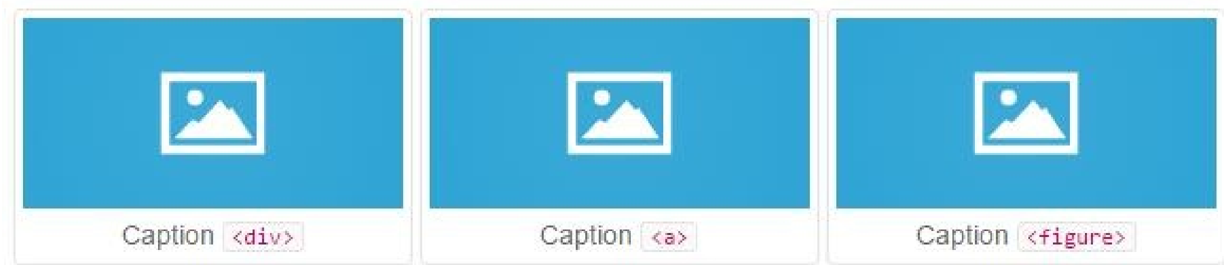
<!-- This is an anchor as a thumbnail -->
<a class="uk-thumbnail" href=""><img src="" alt=""></a>

<!-- This is a figure as a thumbnail -->
<figure class="uk-thumbnail"><img src="" alt=""></figure>
```

图片说明

添加 `.uk-thumbnail-caption` 类到一个 `<div>` 元素中，可以在图片下面添加一个图片说明。

示例



Code

```
<!-- This is a div as a thumbnail with a caption -->
<div class="uk-thumbnail">
    <img src="" alt="">
    <div class="uk-thumbnail-caption">...</div>
</div>

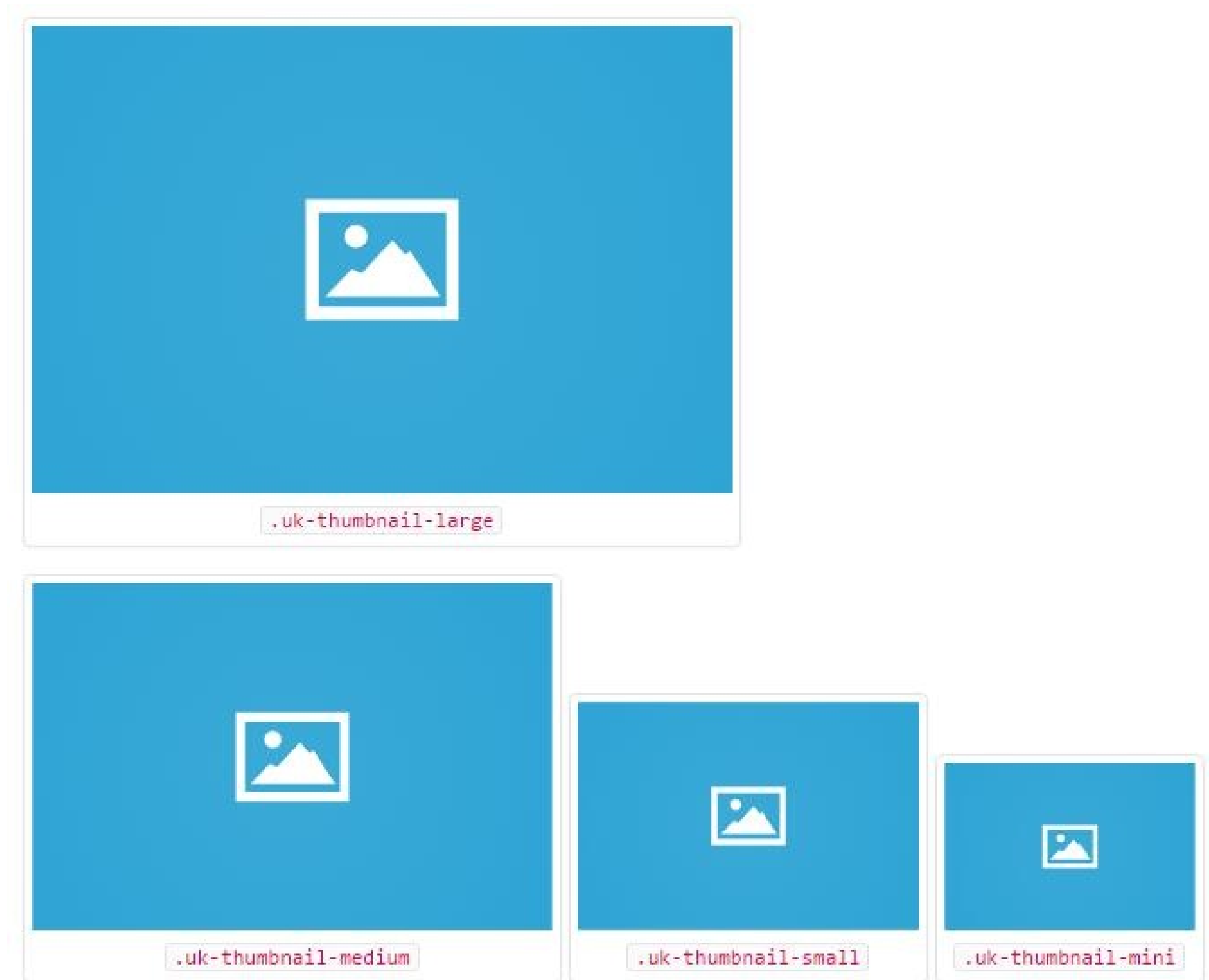
<!-- This is an anchor as a thumbnail with a caption -->
<a class="uk-thumbnail" href="">
    <img src="" alt="">
    <div class="uk-thumbnail-caption">...</div>
</a>

<!-- This is a figure as a thumbnail with a caption -->
<figure class="uk-thumbnail">
    <img src="" alt="">
    <figcaption class="uk-thumbnail-caption">...</figcaption>
</figure>
```

尺寸调整

使用 `.uk-thumbnail-large` , `.uk-thumbnail-medium` , `.uk-thumbnail-small` 或 `.uk-thumbnail-mini` 类来调整图片的尺寸。在 基础组件 中要求图片默认地具有响应式性能。

示例

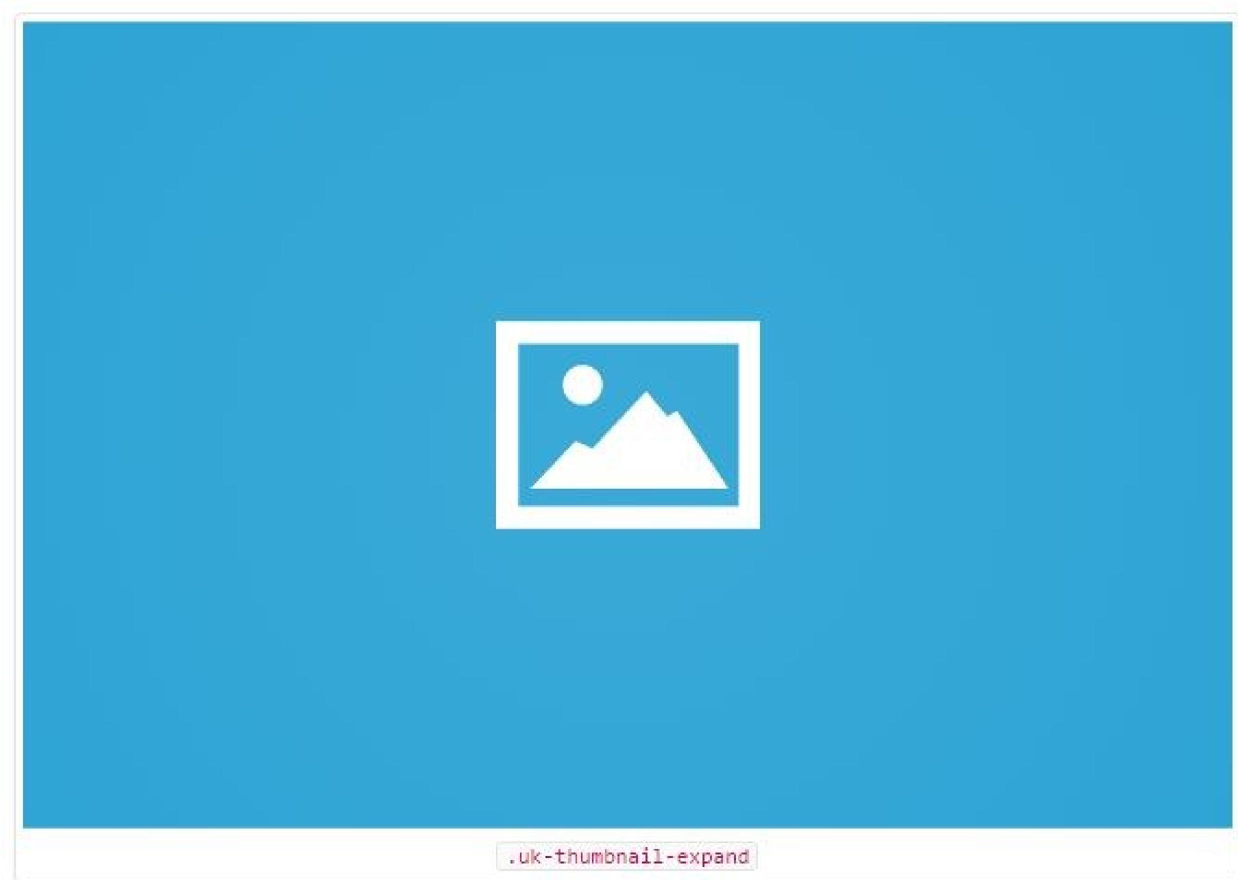


Code

```
<div class="uk-thumbnail uk-thumbnail-large">...</div>
<div class="uk-thumbnail uk-thumbnail-medium">...</div>
<div class="uk-thumbnail uk-thumbnail-small">...</div>
<div class="uk-thumbnail uk-thumbnail-mini">...</div>
```

你甚至可以将缩放缩略图超出其正常大小，使其延伸到它的父元素的宽度。只需添加 `.uk-thumbnail-expand` 类即可。

示例



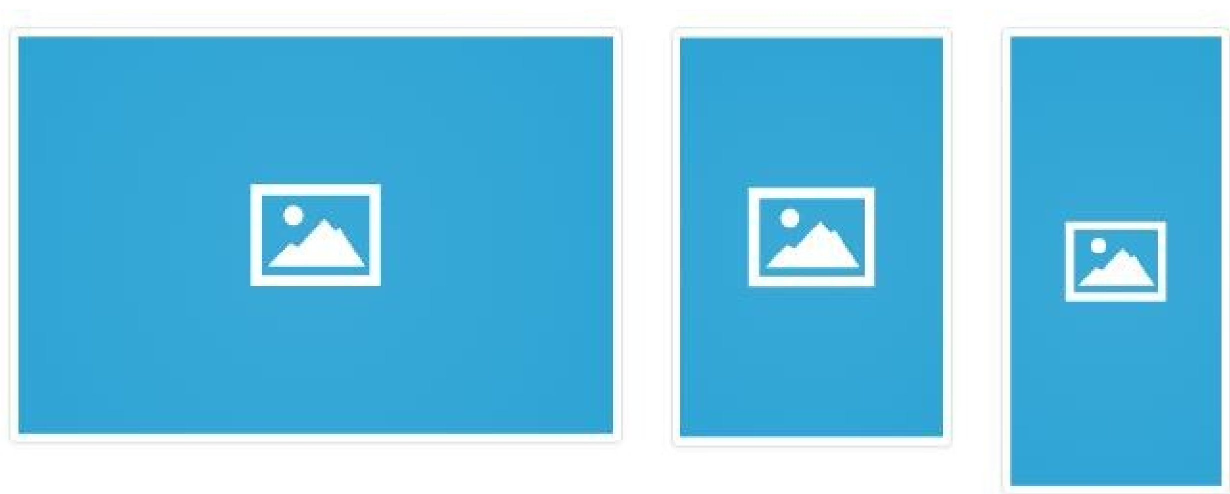
Code

```
<div class="uk-thumbnail uk-thumbnail-expand">...</div>
```

网格

你可以通过使用 网格组件 轻松地创建带有缩略图的网格。

示例



Code

```
<div class="uk-grid">
    <div class="uk-width-medium-1-2"><img class="uk-thumbnail" src=
    <div class="uk-width-medium-1-2">
        <div class="uk-grid">
            <div class="uk-width-medium-1-2"><img class="uk-thumbna
            <div class="uk-width-medium-1-2"><img class="uk-thumbna
        </div>
</div>
```

遮罩/Overlay

创建拥有不同样式的图片遮罩效果。

用法

这个组件使用方法很简单。创建一个定位遮罩层的语境，需要为包含图片的容器元素添加 `.uk-overlay` 类。为一个子元素添加 `.uk-overlay-panel` 类创建真实的遮罩面板。通常你需要使用 `<figure` 和 `<figcaption>` 元素。

Example



Markup

```
<figure class="uk-overlay">
    <img src="" width="" height="" alt="">
    <figcaption class="uk-overlay-panel">...</figcaption>
</figure>
```

通过hover拨动遮罩效果

默认情况下，遮罩层是可见的。添加 `.uk-overlay-hover` 类到遮罩层的容器，实现遮罩层的隐藏和hover拨动效果。

Example




```
<figure class="uk-overlay uk-overlay-hover">
    <img src="" width="" height="" alt="">
    <figcaption class="uk-overlay-panel">...</figcaption>
</figure>
```

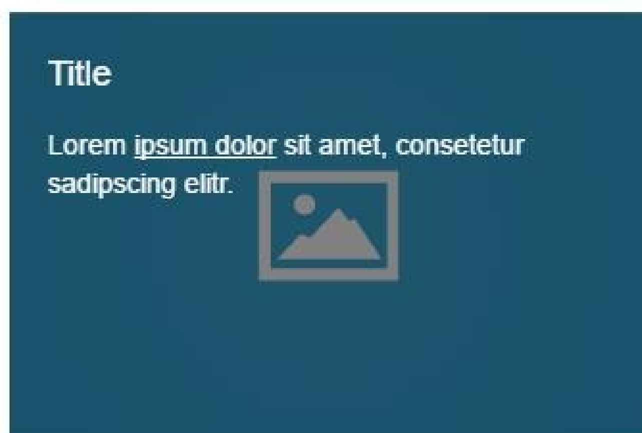
active状态拨动遮罩/Toggle on active

想要在遮罩层的父容器处于actvie状态时拨动遮罩层，添加 `.uk-overlay-active` 类即可。这会在诸如用在幻灯片等情况下带来很多方便。

遮罩背景

为遮罩层的容器添加 `.uk-overlay-background` 类，就能赋予遮罩层背景效果。

Example



Markup

```
<figure class="uk-overlay">
    <img src="" width="" height="" alt="">
    <figcaption class="uk-overlay-panel uk-overlay-background">...<
</figure>
```

遮罩层的图标

基本上可以把任意内容放在遮罩层上面。你也可以添加 `.uk-overlay-icon` 类到遮罩面板中，将会显示一个图标。

Example



Markup

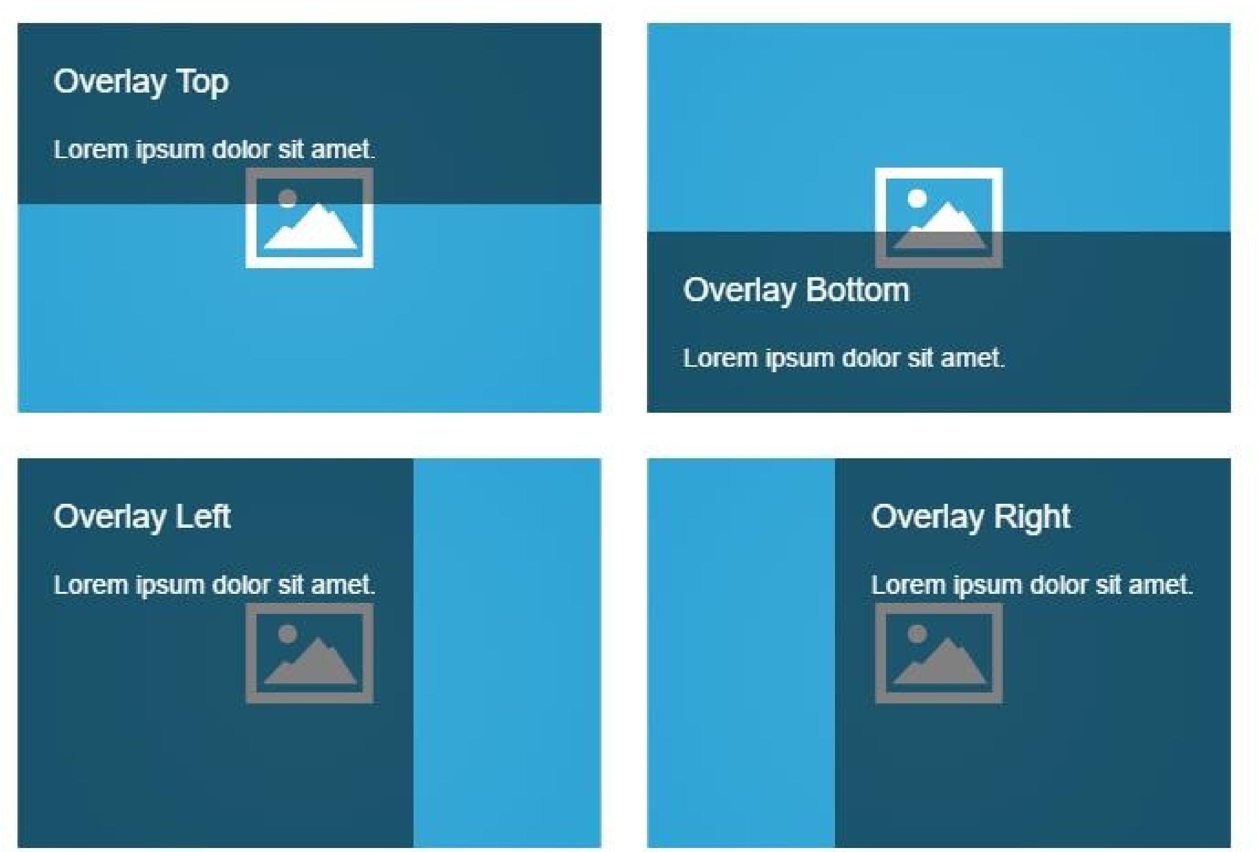
```
<figure class="uk-overlay">
    <img src="" width="" height="" alt="">
    <figcaption class="uk-overlay-panel uk-overlay-icon"></figcapti
</figure>
```

遮罩定位

默认地，遮罩层从左上角开始覆盖整个父容器。要想定位和修剪遮罩层，添加以下类中的一个到遮罩面板中即可。

Class	描述
`.uk-overlay-top`	遮罩层顶部对齐
`.uk-overlay-bottom`	遮罩层底部对齐
`.uk-overlay-left`	遮罩层左对齐
`.uk-overlay-right`	遮罩层右对齐

Example



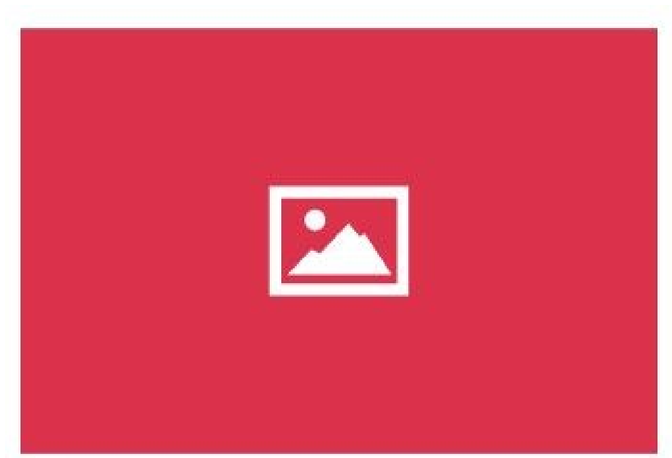
Markup

```
<figure class="uk-overlay">
    <img src="" width="" height="" alt="">
    <figcaption class="uk-overlay-panel uk-overlay-top">...</figcap
</figure>
```

图片遮罩

要使用图片作为遮罩层，需要为带有 `.uk-overlay-panel` 的 `<img>` 元素添加 `.uk-overlay-image` 类名。

Example



Markup

```
<figure class="uk-overlay">
    <img src="" width="" height="" alt="">
    <img class="uk-overlay-panel uk-overlay-image" src="" width=""
</figure>
```

遮罩的过渡效果

遮罩层面板和图片都可以轻易做成动画效果。只需添加以下类中的一个到遮罩层面板或者 `<img>` 元素。记住，这玩意只能与 `.uk-overlay-hover` 或 `.uk-overlay-active` 组合使用。

Class	描述
`.uk-overlay-slide-top`	添加这个类到遮罩面板，使它由顶部滑出。
`.uk-overlay-slide-bottom`	添加这个类到遮罩面板，使它由底部滑出。
`.uk-overlay-slide-left`	添加这个类到遮罩面板，使它由左边滑出。
`.uk-overlay-slide-right`	添加这个类到遮罩面板，使它由右边滑出。
`.uk-overlay-fade`	添加这个类到遮罩面板或图片，使它淡入。
`.uk-overlay-scale`	添加这个类到图片使它放大。
`.uk-overlay-spin`	添加这个类到图片，使它向右轻轻旋转。
`.uk-overlay-grayscale`	添加这个类到图片，hover时去饱和度并着色。

NOTE 如果你想要图片具有动画效果，并希望遮罩面板总是可见的，只需要添加 `.uk-ignore` 类名到面板。这样，它就能无视 `.uk-overlay-hover` 的效果了。

Example



Markup

```
<!-- This example is sliding in the overlay panel from the top -->
<figure class="uk-overlay uk-overlay-hover">
    <img src="" width="" height="" alt="">
    <figcaption class="uk-overlay-panel uk-overlay-slide-top">...</
</figure>

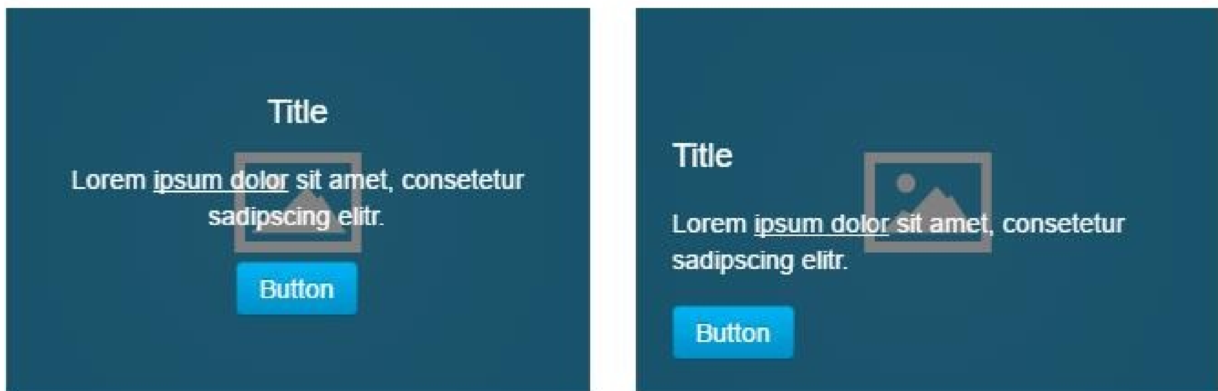
<!-- This example is scaling up the image while the overlay panel a
<figure class="uk-overlay uk-overlay-hover">
    <img class="uk-overlay-scale" src="" width="" height="" alt="">
    <figcaption class="uk-overlay-panel uk-ignore">...</figcaption>
</figure>

<!-- This example is spinning the image and sliding in the overlay
<figure class="uk-overlay uk-overlay-hover">
    <img class="uk-overlay-spin" src="" width="" height="" alt="">
    <figcaption class="uk-overlay-panel uk-overlay-slide-top">...</
</figure>
```

遮罩与 flex

你可以使用 Flex 组件 在垂直和水平方向上居中遮罩面板，而无需进行裁剪。

Example



Markup

```
<!-- In this example the overlay panel is centered vertically and h
<figure class="uk-overlay">
    <img src="" width="" height="" alt="">
    <figcaption class="uk-overlay-panel uk-flex uk-flex-center uk-f
        <div>...</div>
    </figcaption>
</figure>

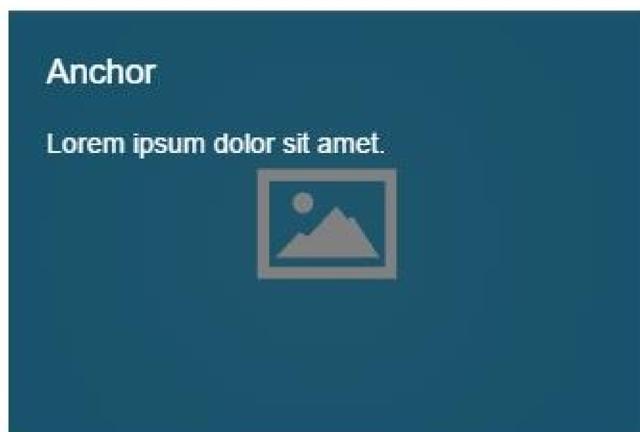
<!-- In this example the overlay panel is aligned to the bottom -->
<figure class="uk-overlay">
    <img src="" width="" height="" alt="">
    <figcaption class="uk-overlay-panel uk-flex uk-flex-bottom">
        <div>...</div>
    </figcaption>
</figure>
```

遮罩层的锚

将整个遮罩层作为链接，只需要将 `<a>` 元素放入到遮罩层容器，并添加来自效果组件 的 `.uk-position-cover` 类名。

重要 应用这个效果时，需要确保已经移除了带有其他容器元素的 `<figcaption>` ，比如 `<div>` 。否则标签不能生效。

Example



Markup

```
<figure class="uk-overlay">
    <img src="" width="" height="" alt="">
    <div class="uk-overlay-panel">...</div>
    <a class="uk-position-cover" href=""></a>
</figure>
```

NOTE 只将遮罩面板作为链接，只需要将锚移至遮罩面板内即可。

文本

实用的内容文本样式类的集合。

文本样式

UIkit提供了各种各样的文本效果用来风格化你的文本。

类	描述
`.uk-text-small`	添加这个类来缩小字体尺寸。
`.uk-text-large`	添加这个类来增大字体尺寸。
`.uk-text-bold`	添加这个类创建粗体文本。
`.uk-text-muted`	添加这个类来淡化文本。
`.uk-text-primary`	为额外的文本信息添加这个类。
`.uk-text-success`	添加这个类来标示成功。
`.uk-text-warning`	添加这个类来标示警告。
`.uk-text-danger`	添加这个类来标示危险。
`.uk-text-contrast`	添加这个类，使文本的色彩在艳丽的色彩或图片背景上更有可读性。它通常会是白色。

文本对齐

使用以下的类来对齐文本

Class	Description
`.uk-text-left`	文本左对齐
`.uk-text-left-small`	只在小屏幕设备上文本左对齐
`.uk-text-left-medium`	在中小型屏幕设备上左对齐
`.uk-text-right`	文本右对齐
`.uk-text-center`	文本水平居中
`.uk-text-center-small`	只在小屏幕设备上文本水平居中
`.uk-text-center-medium`	在中小型屏幕设备上文本水平居中
`.uk-text-justify`	文本两端对齐

Example

Lorem ipsum dolor sit amet, consetetur sadipscing elitr.
`.uk-text-left`

Lorem ipsum dolor sit amet, consetetur sadipscing elitr.
`.uk-text-right`

Lorem ipsum dolor sit amet, consetetur sadipscing elitr.
`.uk-text-center`

Lorem ipsum dolor sit amet, consetetur sadipscing elitr.
`.uk-text-center-small`

Lorem ipsum dolor sit amet, consetetur sadipscing elitr.
`.uk-text-center-medium`

Lorem ipsum dolor sit amet, consetetur sadipscing elitr.
`.uk-text-justify`

垂直对齐

添加这些Class中的一个来使文本与某个对象垂直对齐。

Class	描述
`.uk-text-top`	顶部对齐文本
`.uk-text-middle`	垂直居中文本
`.uk-text-bottom`	底部对其文本

Example



文本换行

添加以下类中的一个到文本的容器，以使文本换行。

Class	描述 Description
`.uk-text-truncate`	通过截断文本，防止文本多行显示
`.uk-text-break`	如果文本长度将超过容器宽度时，强制换行。
`.uk-text-nowrap`	防止文本被截断成多行显示。

Lorem ipsum dolor sit amet, con...

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt. Lorem ipsum dolor sit amet, consetetur sadipscing elit
.uk-text-break

Aenean vulputate eleifend tellus. Aenean leo ligula, porttitor eu, consequat vitae, eleifend ac.
.uk-text-nowrap

列

将文本段落分割成多个列进行显示。

用法

将文本段落分割成多个列，只需要添加一个 `.uk-column-1-*` 类名即可生效。

Class	Description
`.uk-column-1-2`	将文本段落分割成两列
`.uk-column-1-3`	将文本段落分割成三列
`.uk-column-1-4`	将文本段落分割成四列
`.uk-column-1-5`	将文本段落分割成五列
`.uk-column-1-6`	将文本段落分割成六列

Example

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Markup

```
<p class="uk-column-1-3">...</p>
```

响应式布局中的列

列组件通用提供了只在特定视口宽度中生效的响应式类名

Class	Description
`.uk-column-*`	所有设备宽度都生效
`.uk-column-small-*`	只在设备宽度 *480px* 以上时生效
`.uk-column-medium-*`	只在设备宽度 *768px* 以上时生效
`.uk-column-large-*`	只在设备宽度 *960px* 以上时生效
`.uk-column-xlarge-*`	只在设备宽度 *1200px* 以上时生效

Example

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Markup

```
<p class="uk-column-xlarge-1-4 uk-column-large-1-3 uk-column-mediur
```

动画

收集了一些可以流畅运行在你的页面中的简单动画。

用法

要使用这个组件的话，添加任意一个 `.uk-animation-*` 类到一个元素中，它就会以一个炫酷的动画效果淡入。这些类通常是使用JavaScript实现动画的具体行为的。

Class类	描述
`.uk-animation-fade`	元素淡入
`.uk-animation-scale-up`	元素由小变大
`.uk-animation-scale-down`	元素由大变小
`.uk-animation-slide-top`	元素从顶部滑入
`.uk-animation-slide-bottom`	元素从底部滑入
`.uk-animation-slide-left`	元素从左边滑入
`.uk-animation-slide-right`	元素从右边滑入
`.uk-animation-shake`	元素震动
`.uk-animation-scale`	元素由大变小而不改变透明度

Example

点击任意方框观看动画。



Markup

```
<div class="uk-animation-fade">...</div>
```

反向动画/Reverse modifier

默认地，所有动画都是元素出现的效果。要把动画反向，添加 `.uk-animation-reverse` 类就行啦。

Example

点击任意方框观看动画。



Markup

```
<div class="uk-animation-fade uk-animation-reverse">...</div>
```

持续时间修改/Duration modifier

要把动画持续时间延长到10秒，就添加 `.uk-animation-10` 类。

Example



Markup

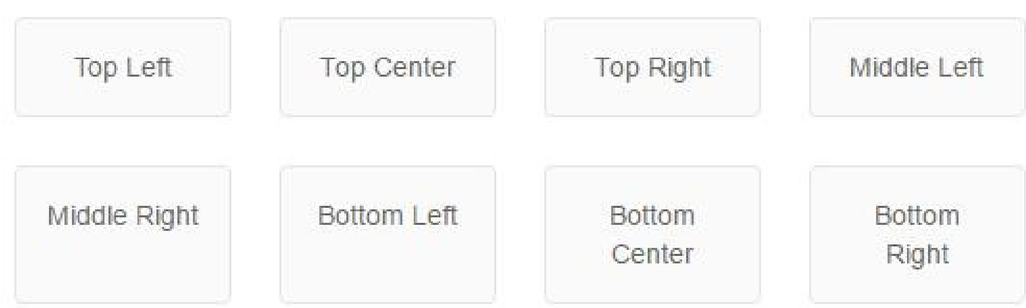
```
<div class="uk-animation-slide-right uk-animation-10">...</div>
```

源头修改/Origin modifier

默认情况下，缩放动画的源头是中间。修改这个行为，添加以下类中的一个就行。

Class类	描述
`.uk-animation-top-left`	动画从左上角展开
`.uk-animation-top-center`	动画从上方中间展开
`.uk-animation-top-right`	动画从右上角展开
`.uk-animation-middle-left`	动画从左边中间展开
`.uk-animation-middle-right`	动画从右边中间展开
`.uk-animation-bottom-left`	动画从左下角展开
`.uk-animation-bottom-center`	动画从底部中间展开
`.uk-animation-bottom-right`	动画从右下角展开

Example



鼠标悬停与动画

通过鼠标悬停触发动画，只需添加 `.uk-animation-hover` 类到动画元素本身或者它的容器中。

Example

Fade Slide right

Markup

```
<div class="uk-animation-hover uk-animation-fade">...</div>

<div class="uk-animation-hover">
    <div class="uk-animation-slide-right">...</div>
</div>
```

对比度/Contrast

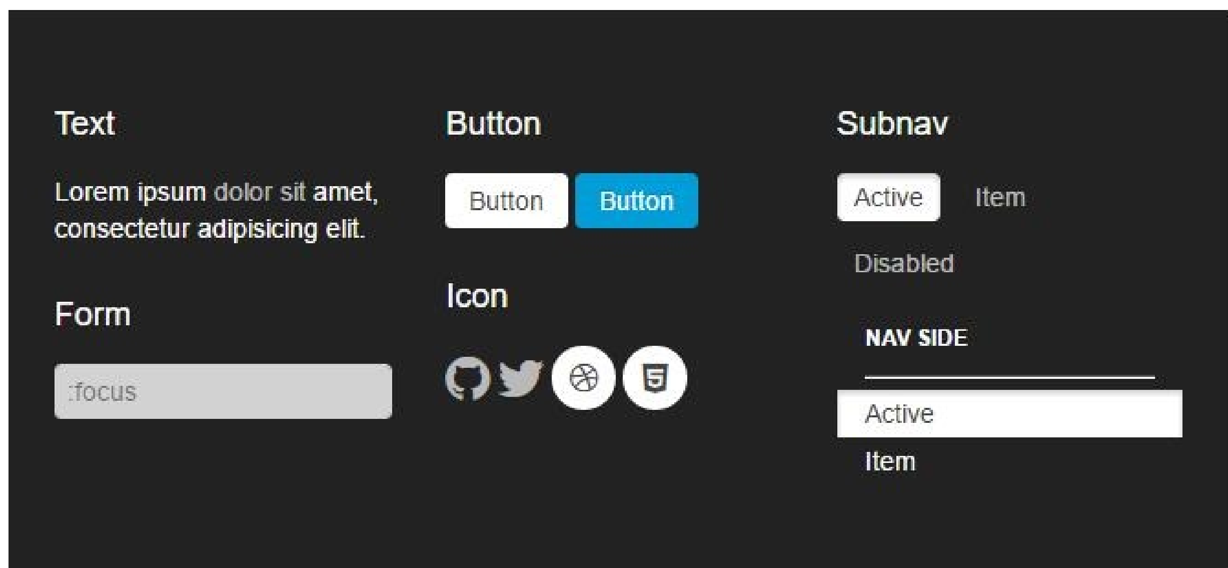
当元素被放置在有色背景或图片上时，使用对比度组件增强元素的外观。

用法

要使用这个组件的话，只需要为容器或者你想要调整其色彩的元素添加 `.uk-contrast` 类。文本，按钮，Nav component 中的侧边导航（Nav Side），子导航组件的分隔线和胶囊效果，图标，表单组件中的默认表单，以及列表组件中的列表线都会自动改变它们的外观。

当内容被放置在块组件中或者放置在图片上时，对比度组件将带来很大的方便。

Example



Markup

```
<div class="uk-contrast">...</div>
```

JAVASCRIPT组件

下拉菜单

为可拨动的下拉菜单定义不同的样式

用法

任意内容，比如一个按钮，都可以作为切换器用来切换下拉菜单。只需用一个带 `data-uk-dropdown` 属性的 `<div>` 元素将其包裹即可。添加 `.uk-dropdown` 类到一个子级的 `<div>` 元素来创建下拉菜单本身。一个下拉菜单可以通过鼠标悬停或者的点击的方式来拨动。

Data 属性	描述
`data-uk-dropdown`	鼠标悬停打开下拉菜单，并添加一个很小的延迟，这样下拉菜单便不会在你停止在拨动器上悬停时立即消失。
`data-uk-dropdown="{mode:'click'}"`	通过点击拨动器来打开下拉菜单，再次点击，下拉菜单便关闭。

重要 为了应用一个下拉菜单，其父元素拥有一个能恰当地将二者对齐的相对位置是很重要的。 很多组件默认地创建了这样的定位场景，比如 按钮，导航栏，二级导航 和 选项卡 等组件。

示例

鼠标来悬停我呀 ▾ 点击我呀 ▾

注意 在这个例子理我

Item
Another item
HEADER
Item
Another item
Separated item

Code

```
<!-- 这是JavaScript关
<div data-uk-dropdown>

    <!-- 拨动下拉菜单,
    <div>...</div>

    <!-- 这是下拉菜单
```

注意 在这个例子理我们使用了 按钮组件 作为拨动器。

Code

```
<!-- 这是JavaScript关联的容器 -->
<div data-uk-dropdown>

    <!-- 拨动下拉菜单的元素 -->
    <div>...</div>

    <!-- 这是下拉菜单 -->
    <div class="uk-dropdown">...</div>

</div>

<!-- 这是关联了点击模式的JavaScript的容器 -->
<div data-uk-dropdown="{mode:'click'}">

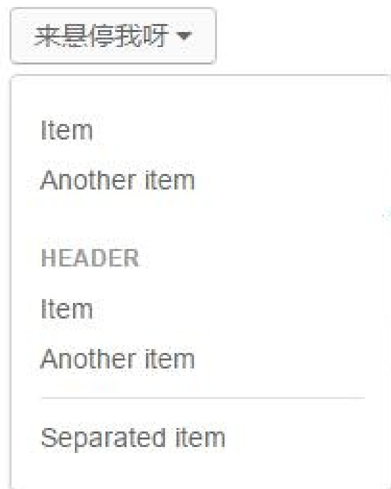
    <!-- 拨动下拉菜单的元素 -->
    <div>...</div>

    <!-- 这是下拉菜单 -->
    <div class="uk-dropdown">...</div>

</div>
```

在悬停模式下延迟下拉菜单

你可以设置一个以毫秒为单位的 `delay` 参数来防止下拉菜单立即显示出来。



Code

```
<div class="uk-dropdown" data-uk-dropdown="{delay: 1000}">
    ...
</div>
```

带有导航菜单的下拉菜单

下拉菜单可以包含一个 导航菜单，只需添加 `.uk-nav` 类和 `.uk-nav-dropdown` 修饰类到一个 `<ul>` 元素中。

Code

```
<div class="uk-dropdown">
    <ul class="uk-nav uk-nav-dropdown">...</ul>
</div>
```

对齐修饰

给下拉菜单添加对齐样式，只需添加以下 `pos:''` 参数中的一个到 data属性就行了。

参数	描述
`pos:'bottom-left'`	默认。下方左对齐
`pos:'bottom-center'`	下方居中对齐
`pos:'bottom-right'`	下方右对齐
`pos:'top-left'`	上方左对齐
`pos:'top-center'`	上方居中对齐
`pos:'top-right'`	上方右对齐
`pos:'left-top'`	左侧顶部对齐
`pos:'left-center'`	左侧垂直居中对齐
`pos:'left-bottom'`	左侧底部对齐
`pos:'right-top'`	右侧顶部对齐
`pos:'right-center'`	右侧垂直居中对齐
`pos:'right-bottom'`	右侧底部对齐

Example



Markup

```
<div data-uk-dropdown="{pos:'bottom-center'}"> ... </div>

<div data-uk-dropdown="{pos:'top-right'}"> ... </div>

<div data-uk-dropdown="{pos:'left-center'}"> ... </div>

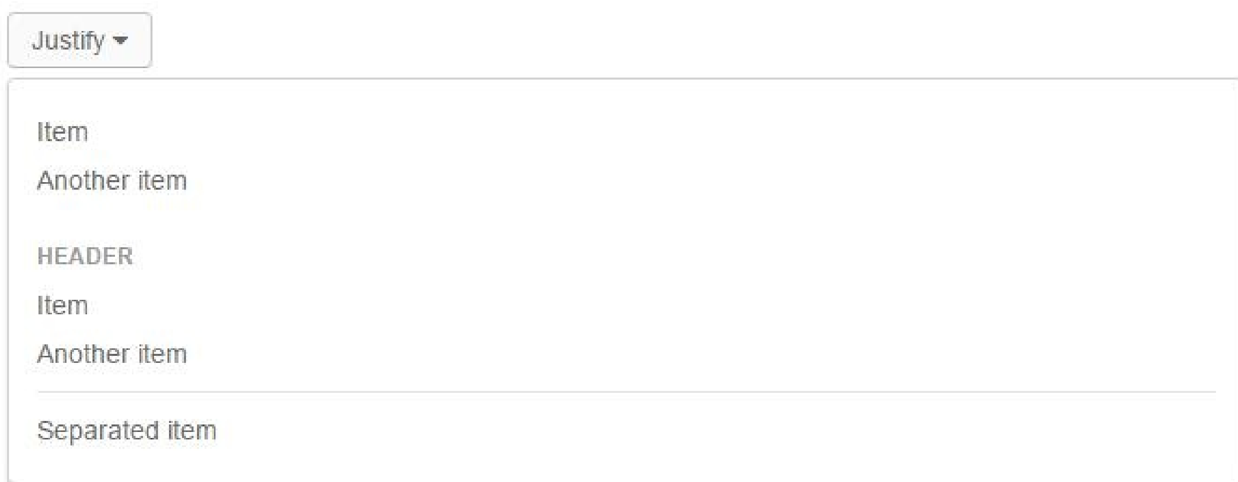
<div data-uk-dropdown="{pos:'right-top'}"> ... </div>
```

调整下拉菜单

调整下拉菜单，只需添加 `data-uk-dropdown="{justify:'#ID'}"` 属性。需要调整的下拉菜单的父元素需要有一个标记id，这样下拉菜单便会扩宽度至这个被标记的元素的宽度。

示例

Justify



Markup

```
<!-- 这是需要调整的下拉菜单的父元素 -->
<div id="my-id">

    <!-- 这是关联了JavaScript的容器 -->
    <div class="uk-button-dropdown" data-uk-dropdown="{justify:'#my

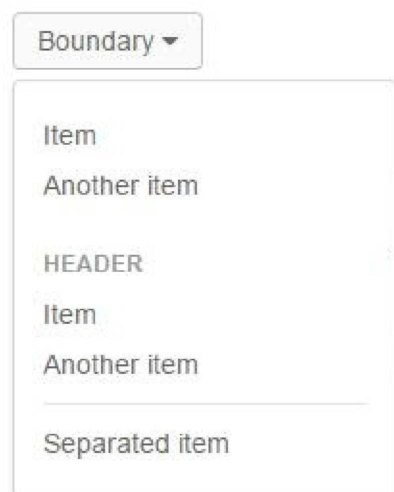
        <!-- 拨动下拉菜单的元素 -->
        <button class="uk-button">...</button>

        <!-- 这是下拉菜单 -->
        <div class="uk-dropdown">...</div>
    </div>
</div>
```

下拉菜单自动翻转

默认情况下，当下拉菜单超出了视口边缘便会自动地翻转。如果你想根据容器的边界来翻转它，只需添加 `data-uk-dropdown="{boundary:'#ID'}"` 属性，这个属性中的ID对应容器的ID。

示例



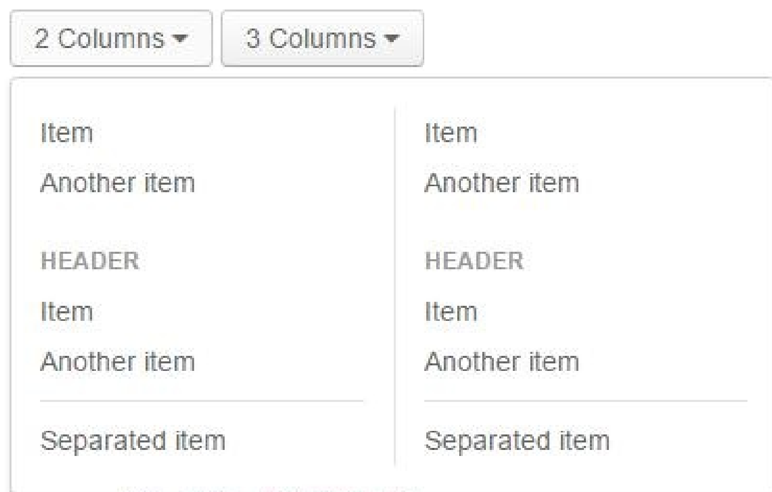
Code

```
<div id="my-id">
    <div class="uk-button-dropdown" data-uk-dropdown="{boundary:'#r
</div>
```

网格

你甚至可以在下拉菜单中放置一个包含导航或者其它内容的 网格组件。只需用一个带有 `.uk-grid` 类的 `<div>` 元素包裹着这些内容。为了优化下拉菜单中的网格，需要添加 `.uk-dropdown-grid` 类名。添加一个 `.uk-dropdown-width-*` 类名到网格的子元素，最多可以并列5个列。

Example



Lorem ipsum dolor sit amet, consectetur elit, sed do eiusmod tempor incididunt.Lorem ipsum dolor sit amet, consectetur elit, sed do eiusmod tempor incididunt.Lorem ipsum dolor sit amet, consectetur elit, sed do eiusmod tempor incididunt.

Markup

```
<div class="uk-dropdown uk-dropdown-width-2">

    <div class="uk-grid uk-dropdown-grid">
        <div class="uk-width-1-2">
            <ul class="uk-nav uk-nav-dropdown uk-panel">...</ul>
            <div class="uk-panel">...</div>
        </div>

        <div class="uk-width-1-2">
            <div class="uk-panel">...</div>
        </div>
    </div>

</div>
```

响应式行为

在狭窄的视口中，比如手机上，可能没有足够的空间来扩展下拉菜单。在这种情况下，下拉菜单会翻转它的对齐准线。如果还是没有足够的空间，网格列会占满宽度并在下拉菜单中垂直地堆叠。

缩小调整

默认情况下，下拉菜单有一个固定的宽度并且文字会切换到下一行。如果你想要你的下拉菜单更小巧，以使它延伸到内容的宽度而不再使文本换行，添加 `.uk-dropdown-small` 类即可。这对一个按钮下拉菜单是很有用的。

示例



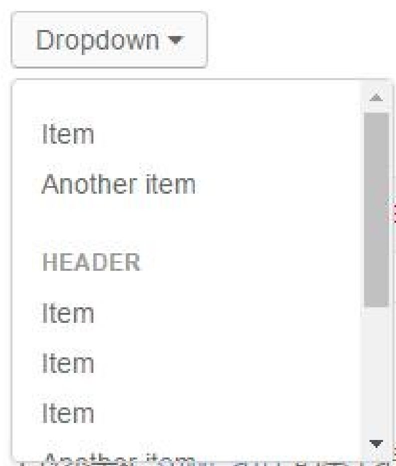
Code

```
<div class="uk-dropdown uk-dropdown-small">...</div>
```

可滚动的修饰

给下拉菜单添加固定高度，使它的内容可以滚动，只需添加 `.uk-dropdown-scrollable` 类。

示例



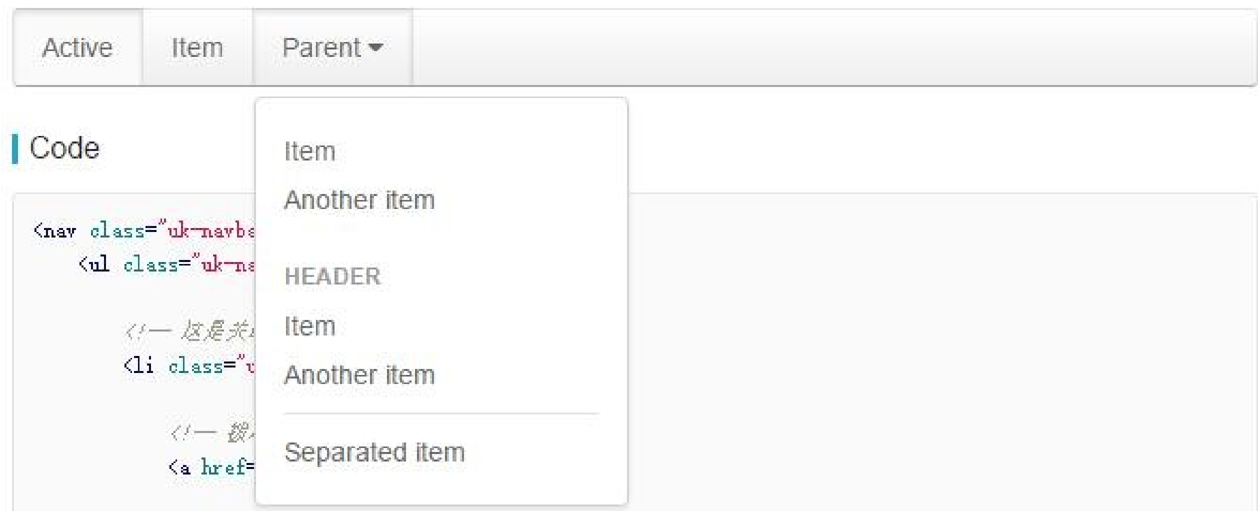
Code

```
<div class="uk-dropdown uk-dropdown-scrollable">...</div>
```

导航栏中的修饰

下拉菜单是 导航栏组件 的基本组成部分。只需添加 `.uk-dropdown-navbar` 类到下拉菜单中，这样下拉菜单便会完美地融入导航栏中。

示例



Code

```
<nav class="uk-navbar">
    <ul class="uk-navbar-nav">

        <!-- 这是关联了JavaScript的容器 -->
        <li class="uk-parent" data-uk-dropdown>

            <!-- 拨动下拉菜单的菜单项 -->
            <a href="">...</a>

            <!-- 这是下拉菜单 -->
            <div class="uk-dropdown uk-dropdown-navbar">
                <ul class="uk-nav uk-nav-navbar">
                    <li><a href="">...</a></li>
                </ul>
            </div>

        </li>
    </ul>
</nav>
```

空白下拉菜单

有时你需要用到下拉菜单的功能但不想要它的样式。这时你直接用 `.uk-dropdown-blank` 来替代 `.uk-dropdown` 就行了。

Example



Title

Lorem ipsum dolor sit amet, consectetur adipisicing elit.

NOTE 在这个例子中，我们使用了面板组件来设定下拉菜单的样式。

Markup

```
<div class="uk-dropdown-blank uk-panel uk-panel-box">...</div>
```

按钮中的下拉菜单

按钮组件 中的按钮可以用来切换下拉菜单。

示例



Code

```
<!-- 这是关联了JavaScript的容器 -->
<div class="uk-button-dropdown" data-uk-dropdown>

    <!-- 这是切换下拉菜单的按钮 -->
    <button class="uk-button">...</button>

    <!-- 这是下拉菜单 -->
    <div class="uk-dropdown uk-dropdown-small">
        <ul class="uk-nav uk-nav-dropdown">
            <li><a href="">...</a></li>
            <li><a href="">...</a></li>
        </ul>
    </div>

</div>
```

按钮组中的下拉菜单

使用 按钮组件 中的按钮组，将按钮分作一个标准的按钮和一个下拉菜单拨动器。

示例



Code

```
<div class="uk-button-group">

    <!-- 这是一个按钮 -->
    <button class="uk-button">...</button>

    <!-- 这是关联了JavaScript的容器 -->
    <div data-uk-dropdown="{mode:'click'}">

        <!-- 拨动下拉菜单的按钮 -->
        <a href="" class="uk-button">...</a>

        <!-- 这是下拉菜单 -->
        <div class="uk-dropdown uk-dropdown-small">
            <ul class="uk-nav uk-nav-dropdown">
                <li><a href="">...</a></li>
                <li><a href="">...</a></li>
            </ul>
        </div>

    </div>
</div>
```

子导航中的下拉菜单

下拉菜单也可以用在 子导航组件 中。

示例



Code

```
<ul class="uk-subnav uk-subnav-pill">
    <li><a href="">...</a></li>

    <!-- 这是关联了JavaScript的容器 -->
    <li data-uk-dropdown="{mode:'click'}">

        <!-- 拨动下拉菜单的导航元素 -->
        <a href="">...</a>

        <!-- 这是下拉菜单 -->
        <div class="uk-dropdown uk-dropdown-small">
            <ul class="uk-nav uk-nav-dropdown">
                <li><a href="">...</a></li>
            </ul>
        </div>

    </li>
</ul>
```

选项卡中的下拉菜单

下拉菜单可以用在 选项卡组件 中。

示例



Code

```
<ul class="uk-tab" data-uk-tab>
    <li><a href="">...</a></li>

    <!-- 这是关联了JavaScript的容器 -->
    <li data-uk-dropdown="{mode:'click'}">

        <!-- 拨动下拉菜单的选项卡标签 -->
        <a href="">...</a>

        <!-- 这是下拉菜单 -->
        <div class="uk-dropdown uk-dropdown-small">
            <ul class="uk-nav uk-nav-dropdown">
                <li><a href="">...</a></li>
            </ul>
        </div>
    </li>
</ul>
```

JavaScript 选项

这是关于如何通过属性设置选项的示例：

```
data-uk-dropdown="{mode:'hover'}"
```

选项	可用的值	默认值	描述
`pos`	string	bottom-left	下拉菜单定位
`mode`	hover, click	hover	触发下拉菜单的行为
`remaintime`	integer	800	悬停模式下，自动关闭下拉菜单前的等待时间
`justify`	CSS selector	false	按指定元素的宽度拉伸下拉菜单
`boundary`	window	CSS 选择器	用被标注的元素来保持下拉菜单的可见性
`delay`	integer	0	悬停模式下，下拉菜单被显示前的延迟时间，以毫秒为单位。
`hoverDelayIdle`	integer	250	以毫秒为单位的，从一个打开下拉菜单将鼠标移至另一个下拉菜单这一过程中的延时。
`preventflip`	mixed	false	防止下拉菜单自动翻转

事件

你可以为自定义的功能绑定回调事件。

事件名称	描述
`show.uk.dropdown`	下拉菜单显示时触发
`hide.uk.dropdown`	下拉菜单隐藏时触发
`stack.uk.dropdown`	当下拉菜单堆叠以适应屏幕时触发

Example

```
$('[data-uk-dropdown]').on('show.uk.dropdown', function(){
    // custom code...
});
```

CSS 选项

添加类名 `.uk-dropdown-close` 到下拉菜单容器或菜单条目，用于点击该条目时隐藏下拉菜单to dropdown container or to item to hide dropdown when user click on item.

模态对话框

创建拥有不同样式和过渡形式的模态对话框。

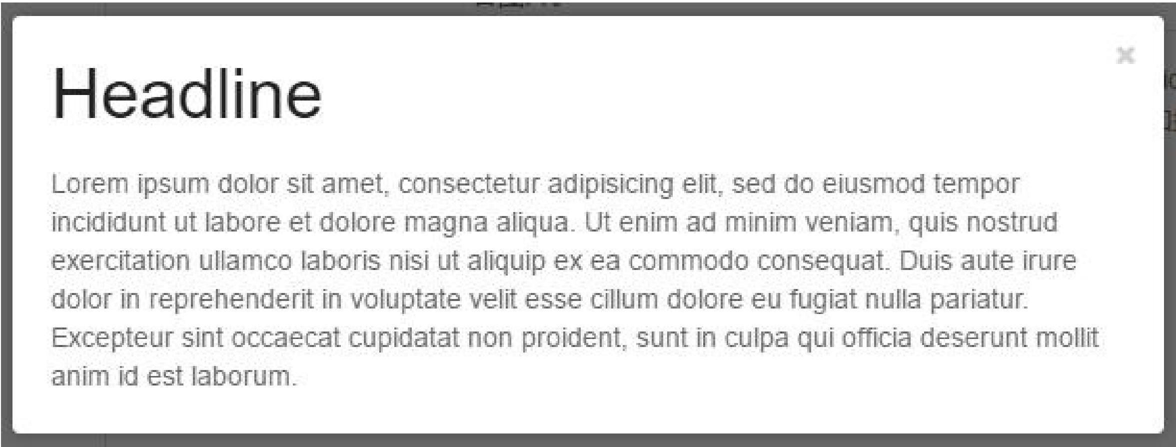
用法

模态对话框由一个遮罩层，一个对话框和一个关闭按钮组成。

Class类	描述
`.uk-modal`	添加这个类到一个 <div> 元素中创建对话框的容器和一个覆盖在页面上的空白遮罩层。为了能拨动这个元素，必须为它添加id。
`.uk-modal-dialog`	为一个后代 <div> 元素添加这个类，创建对话框。
`.uk-modal-close`	添加这个类到 <a> 或 <button> 元素，用来创建对话框中的关闭按钮。推荐为按钮添加关闭按钮组件中的 `.uk-close` 类，以赋予适当的样式，你也可以使用文字或者图片。

你可以使用任意元素来触发模态对话框。一个 <a> 元素需要链接到模态对话框的id 。为了使必要的JavaScript生效，需要添加 `data-uk-modal` 属性。如果你使用的是其他元素，比如按钮，只需要添加 `data-uk-modal="{target:'#ID'}"` 元素指向模态对话框的。

Example



Headline

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Markup

```
<!-- 触发模态对话框的锚 -->
<a href="#my-id" data-uk-modal>...</a>

<!-- 触发模态对话框的按钮 -->
<button class="uk-button" data-uk-modal="{target:'#my-id'}">...</bu

<!-- 模态对话框 -->
<div id="my-id" class="uk-modal">
    <div class="uk-modal-dialog">
        <a class="uk-modal-close uk-close"></a>
        ...
    </div>
</div>
```

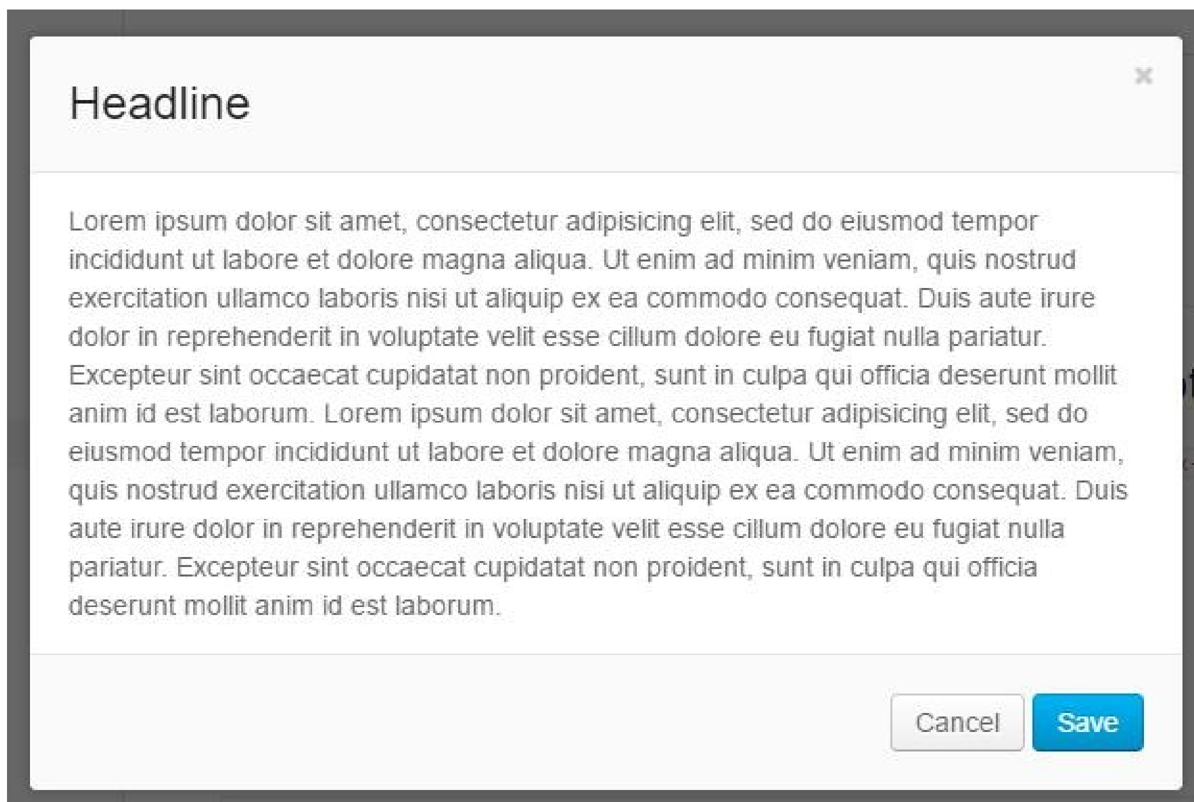
JavaScript 选项

默认地，点击遮罩层会自动关闭模态对话框。如果要阻止它，只需要添加 `data-uk-modal="{target:'#ID',bgclose:false}"` 属性。

模态对话框的标题和脚注/Modal header and footer

你可以为模态对话框创建标题和脚注，使它们与正文内容分离开。只需要添加 `.uk-modal-header` 或者 `.uk-modal-footer` 类到模态对话内部的 `<div>` 元素即可。

Example



Markup

```
<div class="uk-modal">
    <div class="uk-modal-dialog">
        <div class="uk-modal-header">...</div>
        ...
        <div class="uk-modal-footer">...</div>
    </div>
</div>
```

模态对话框的说明文字/Modal caption

你还可以为模态对话框创建一个显示在它外部的说明文字。只需要天 `.uk-modal-caption` 类到模态对话框内部的 `<div>` 元素即可。

Example



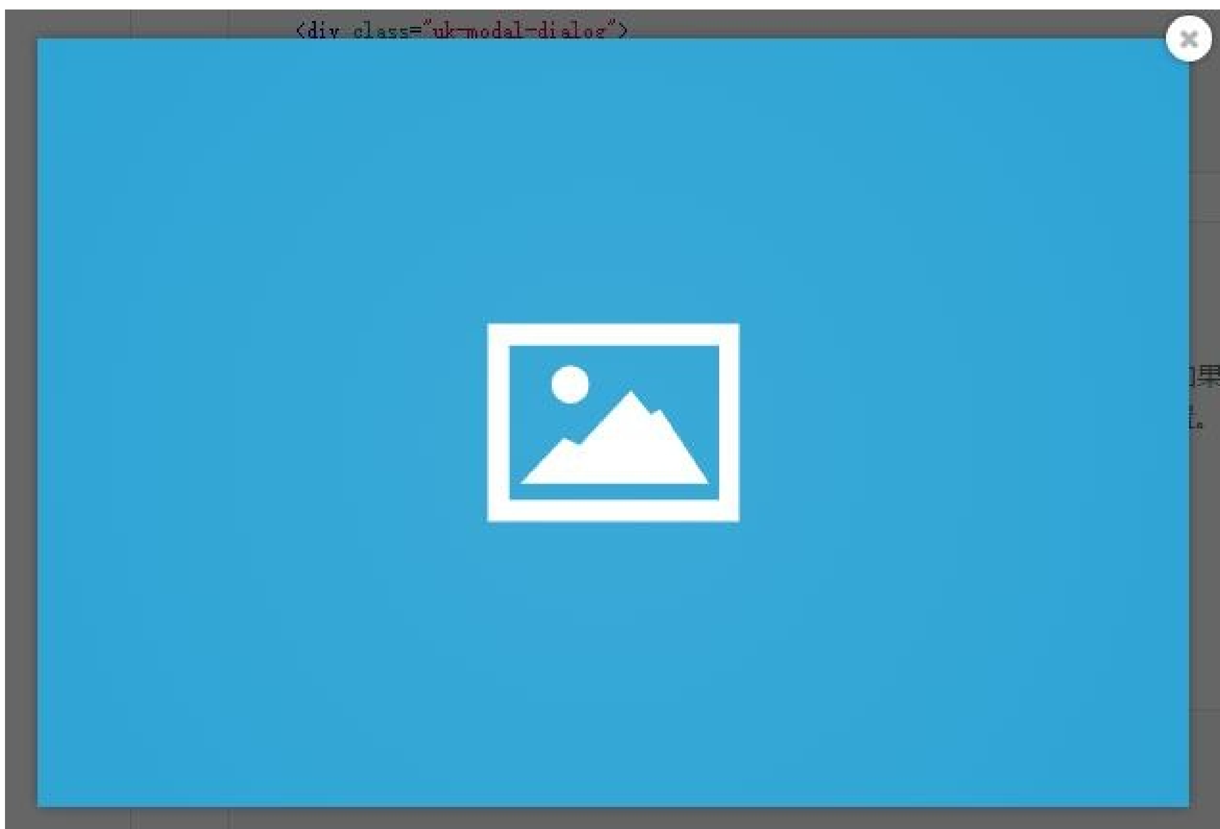
Markup

```
<div class="uk-modal">
    <div class="uk-modal-dialog">
        <div class="uk-modal-caption">...</div>
    </div>
</div>
```

灯箱修饰/Lightbox modifier

创建看起来像灯箱的模态对话框，只需要添加 `.uk-modal-dialog-lightbox` 类。如果你想要把图片显示在灯箱一样的模态对话框中时，这会很有用。关闭按钮会自动调整到对话框的相应位置。

Example



Markup

```
<!-- 触发模态对话框的锚 -->
<a href="#my-id" data-uk-modal>...</a>

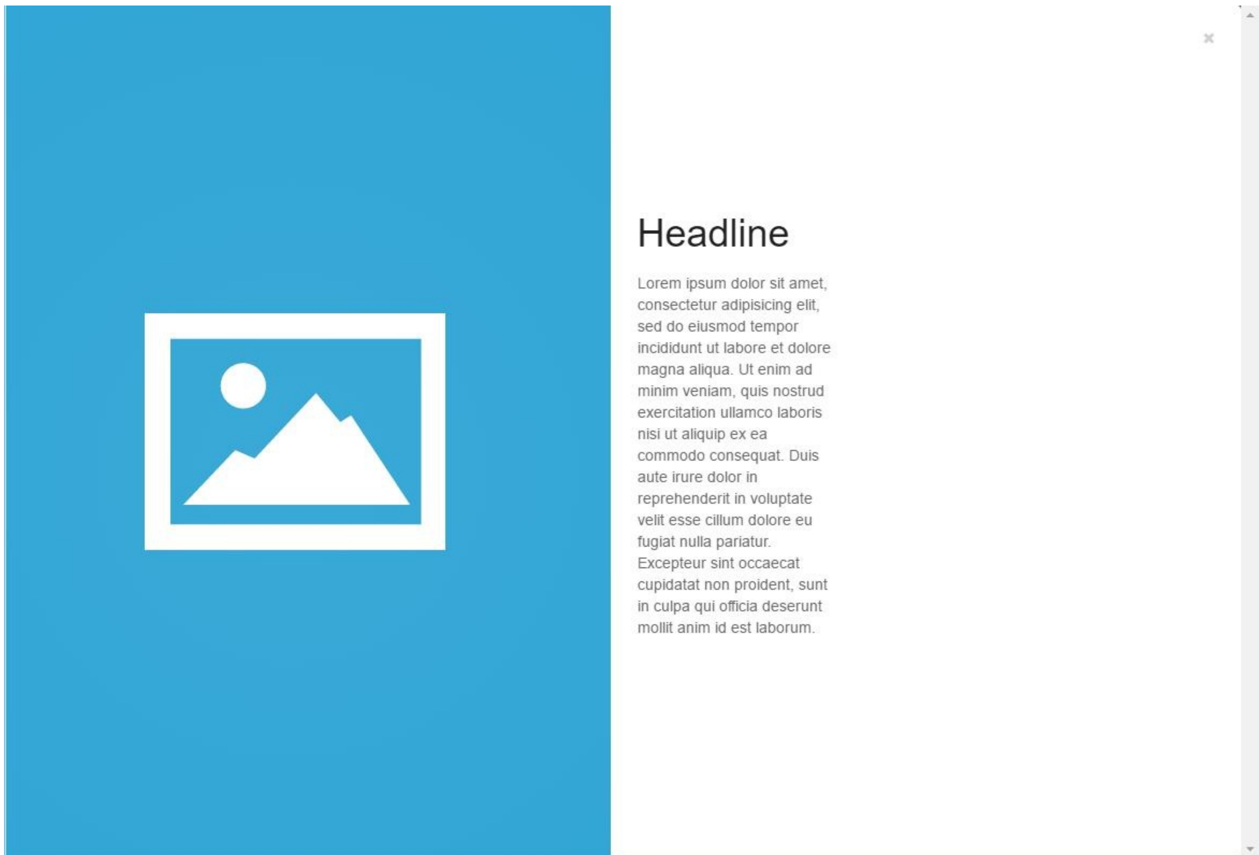
<!-- 模态对话框 -->
<div id="my-id" class="uk-modal">
    <div class="uk-modal-dialog uk-modal-dialog-lightbox">
        <a href="" class="uk-modal-close uk-close uk-close-alt"></a
        <img src="" alt="">
    </div>
</div>
```

注意 在灯箱模态对话框中创建关闭按钮时，我们依然推荐添加 关闭组件 中的 `.uk-close-alt` 类来赋予你的关闭按钮一个适当的样式。

空白模态框

重置所有样式，比如padding和margin,添加 `.uk-modal-dialog-blank` 类名即可。如果你想创建全屏模态框时，可以使用它。此时，你还需要用到效果组件中的 `.uk-height-viewport` 类名，使得模态框填充整个视口高度。

Example




```
<!-- This is the anchor toggling the modal -->
<a href="#my-id" data-uk-modal>...</a>

<!-- This is the modal -->
<div id="my-id" class="uk-modal">
    <div class="uk-modal-dialog uk-modal-dialog-blank">...</div>
</div>
```

模态对话框中的旋转/Modal spinner

要在模态对话框中放入一个旋转的图标，添加 `.uk-modal-spinner` 类到模态对话框中的 `<div>` 元素即可。

Example



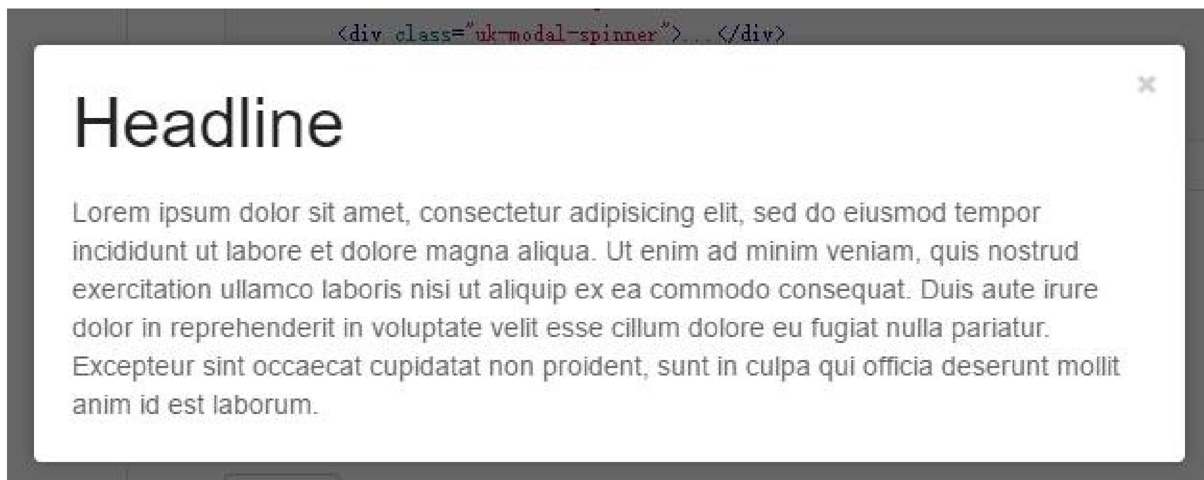
Markup

```
<div class="uk-modal">
    <div class="uk-modal-dialog">
        <div class="uk-modal-spinner">...</div>
    </div>
</div>
```

居中模态对话框

垂直居中模态对话框，添加 `{center:true}` 选项到它的 data属性即可。

Example



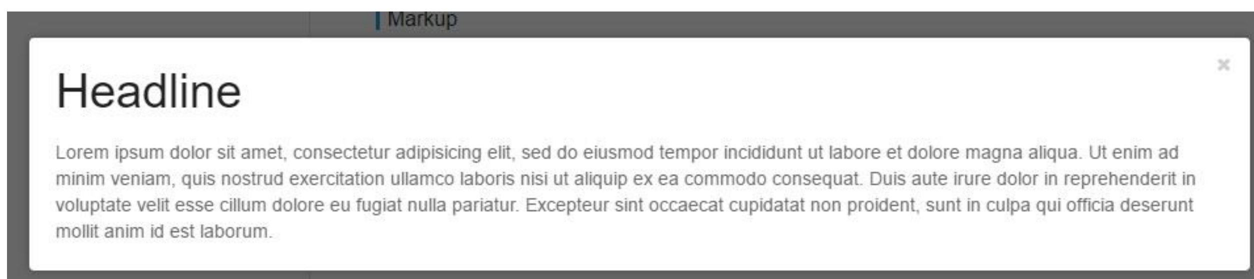
Markup

```
<a href="#my-id" data-uk-modal="{center:true}"></a>
```

大对话框修饰

为模态对话框添加与网页相同的宽度，只需要添加 `.uk-modal-dialog-large` 类。

Example



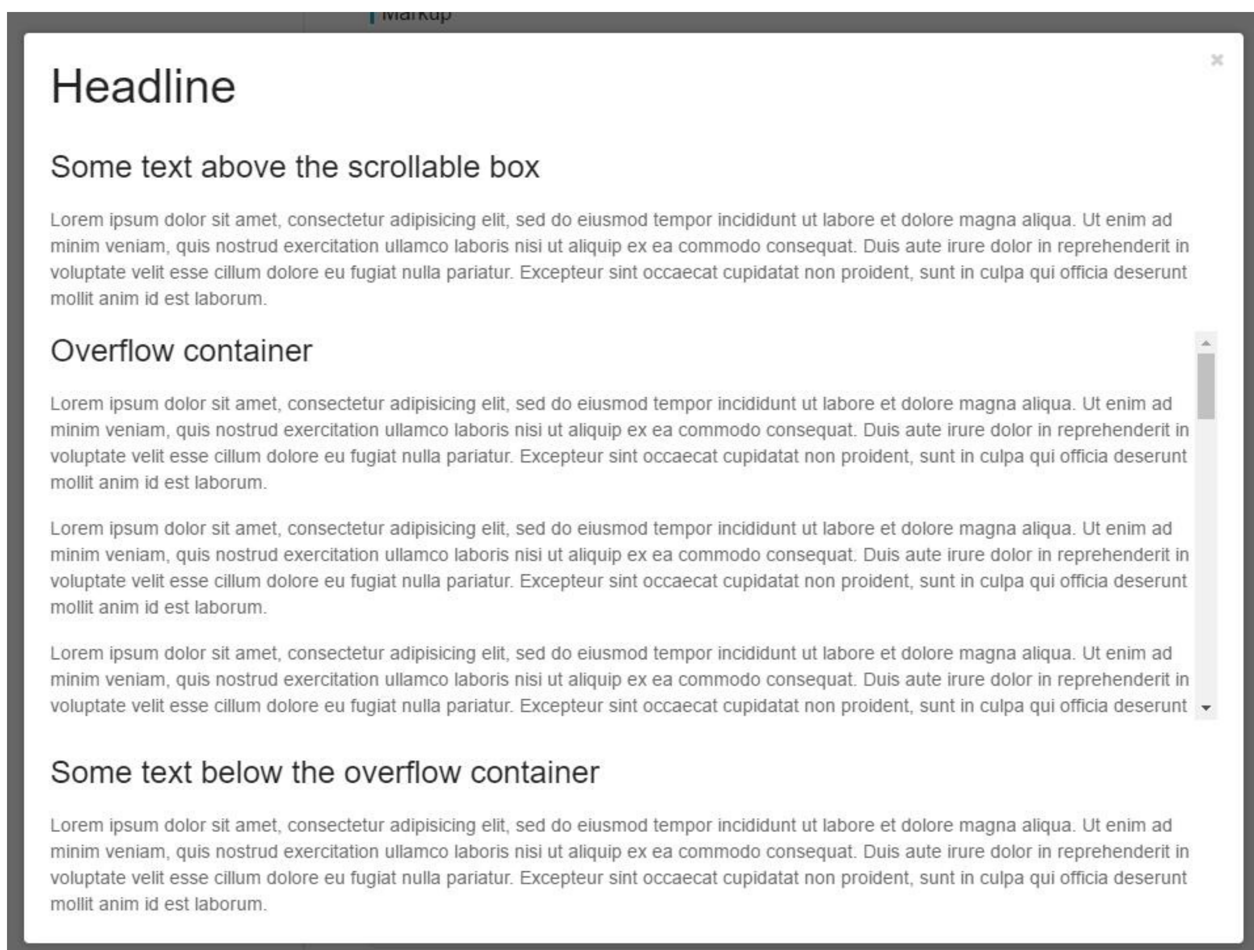
Markup

```
<div class="uk-modal-dialog uk-modal-dialog-large">...</div>
```

模态对话框中溢出容器

你还可以把模态对话框的内容显示在可滚动的容器中。只需添加 `.uk-overflow-container` 类到模态对话框中的 `<div>` 元素即可。模态对话框会自动扩展填充到页面的高度。

Example



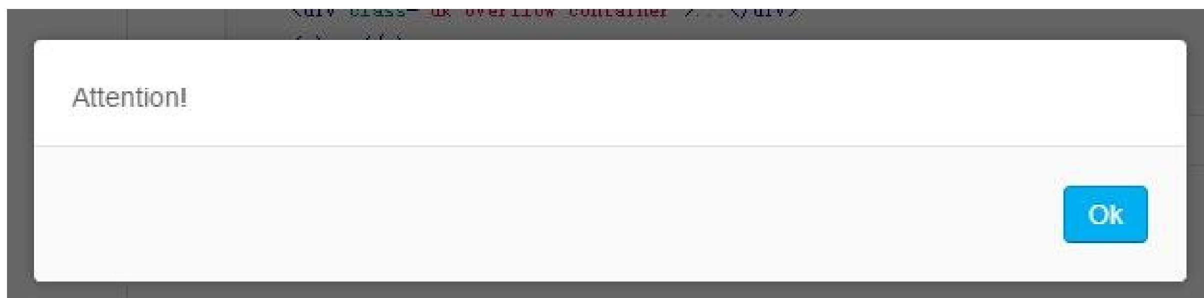
Markup

```
<div class="uk-modal-dialog">
    <p>...</p>
    <div class="uk-overflow-container">...</div>
    <p>...</p>
</div>
```

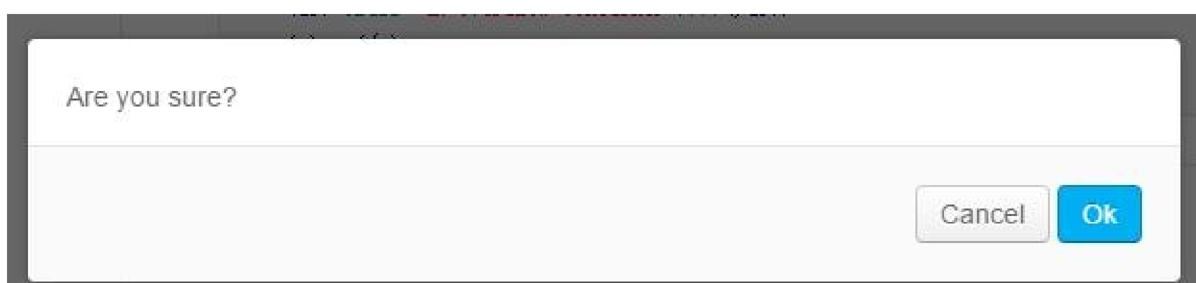
对话框

模态对话框组件还提供了原生对话框的选择： window.alert, window.confirm 和 window.prompt.

Example



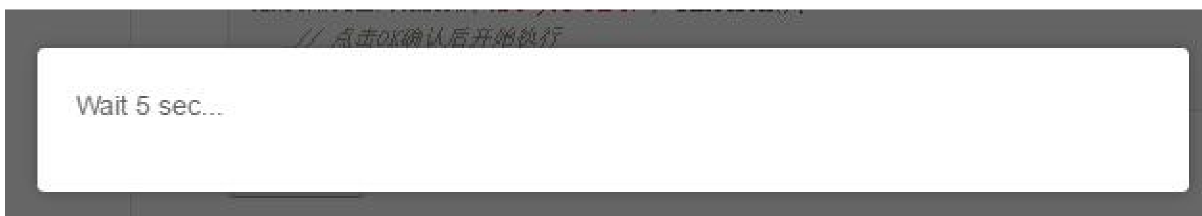

```
UIkit.modal.alert("Attention!");
```



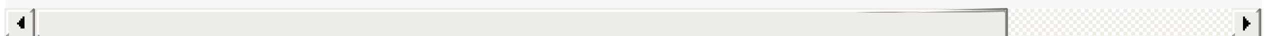

```
UIkit.modal.confirm("Are you sure?", function(){
    // 点击OK确认后开始执行
});
```




```
UIkit.modal.prompt("Name:", value, function(newvalue){
    // 点击OK提交后执行
});
```




```
var modal = UIkit.modal.blockUI("Any content..."); // 使用 modal.hi
```



JavaScript

你可以通过原生的Javascript处理模态对话框。

Example

```
var modal = UIkit.modal(".modalSelector");

if ( modal.isActive() ) {
    modal.hide();
} else {
    modal.show();
}
```

事件

每当模态对话框被打开时会触发一次 `show.uk.modal` 事件，每当被关闭时会触发一次 `hide.uk.modal` 事件。

Example

```
$('.modalSelector').on({

    'show.uk.modal': function(){
        console.log("Modal is visible.");
    },

    'hide.uk.modal': function(){
        console.log("Element is not visible.");
    }
});
```

事件

名称	参数	描述
`show.uk.modal`	event	模态对话框显示时
`hide.uk.modal`	event	模态对话框隐藏时。

抽屉/Off-canvas

创建一个可以在页面上平滑地滑入滑出的抽屉。

抽屉完美适用于构建移动端导航，与那些颇受欢迎的许多原生手机应用类似，在其左上角用一个按钮来开关带有菜单的侧边栏。

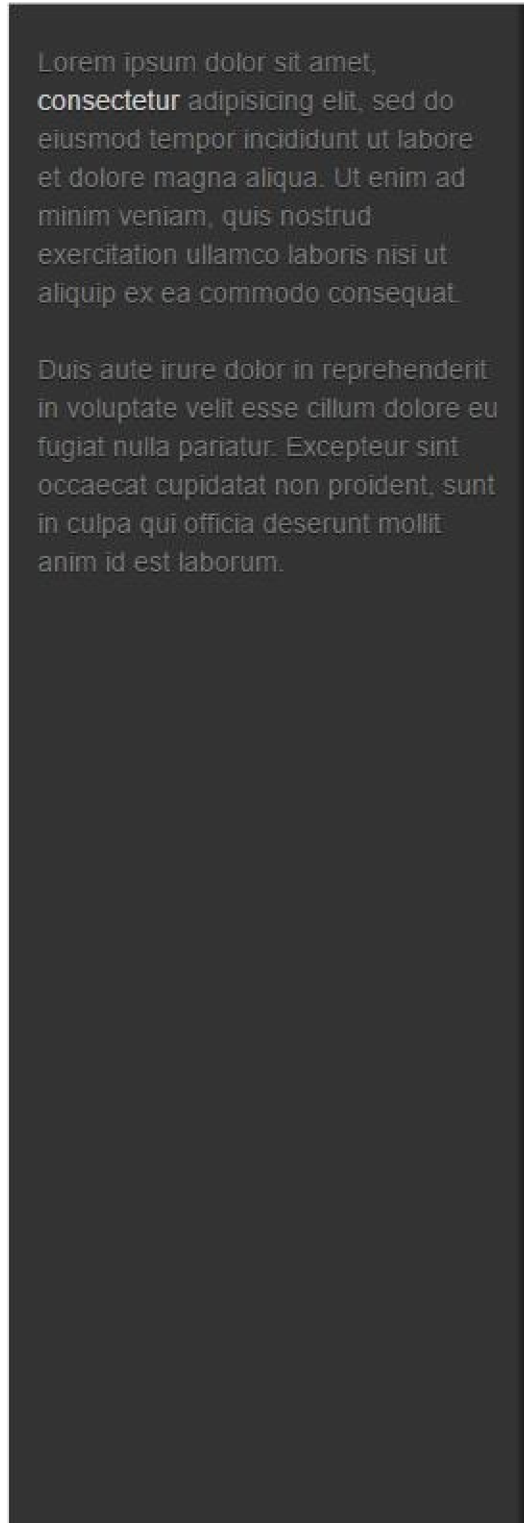
用法

抽屉组件由一个遮罩层和一个弹出边栏组成。

Class	描述
`.uk-offcanvas`	添加该类至一个 `<div>` 元素来创建隐藏在页面外的边栏容器和覆盖层。id也需要添加，使抽屉可被打开或关闭。
`.uk-offcanvas-bar`	添加该类至一个子级 `<div>` 元素来创建抽屉式边栏。

你可以使用任何元素来切换抽屉式侧边栏。 `<a>` 元素需要链接至抽屉容器的id。为了使必要的JavaScript生效，仅需添加 `data-uk-offcanvas` 属性即可。如果你使用了其他元素，比如按钮，仅需添加 `data-uk-offcanvas="{target:'#ID'}"` 属性指向抽屉容器的id。

示例

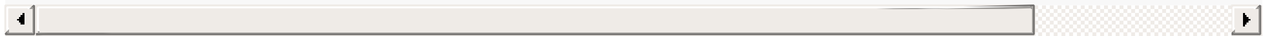

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.
Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Code

```
<!-- 这是开关抽屉式边栏的锚 -->
<a href="#my-id" data-uk-offcanvas>...</a>

<!-- 这是开关抽屉式边栏的按钮 -->
<button class="uk-button" data-uk-offcanvas="{target:'#my-id'}">...

<!-- 抽屉式边栏 -->
<div id="my-id" class="uk-offcanvas">
    <div class="uk-offcanvas-bar">...</div>
</div>
```



翻转修饰

添加 `.uk-offcanvas-bar-flip` 类至抽屉式边栏，使之从右侧滑出。

示例

Lorem ipsum dolor sit amet, **consectetur** adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

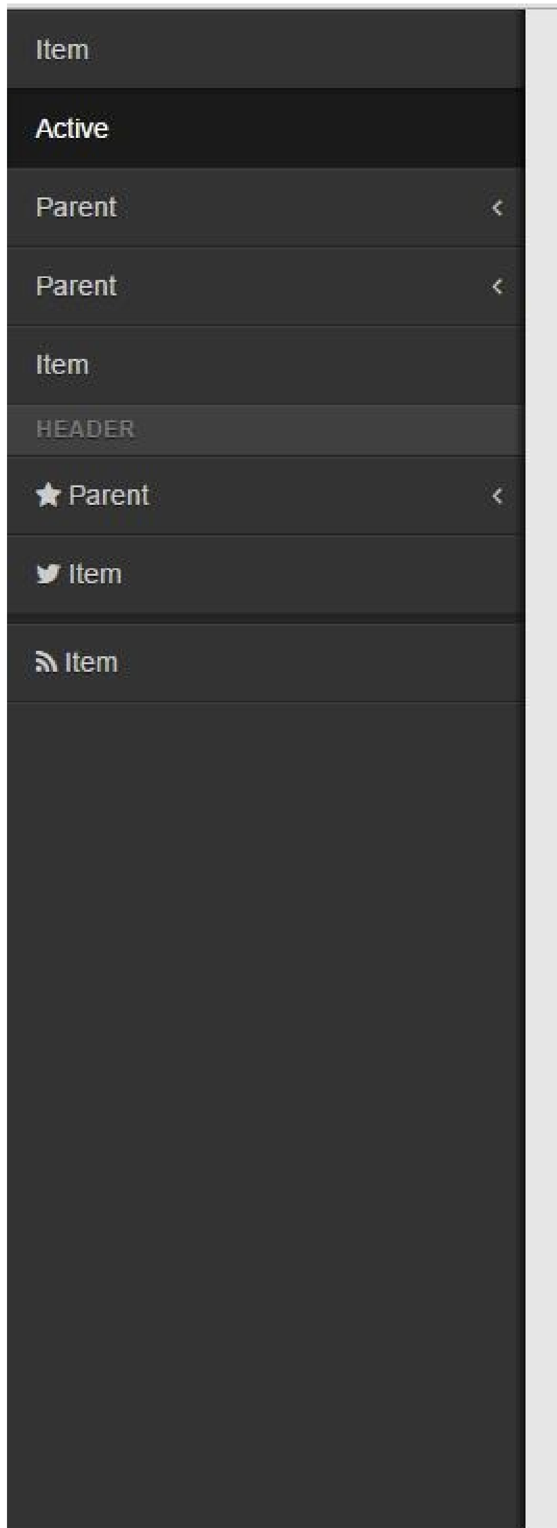
Code

```
<div id="my-id" class="uk-offcanvas">
    <div class="uk-offcanvas-bar uk-offcanvas-bar-flip">...</div>
</div>
```

抽屉式导航

抽屉式侧边栏可以包含导航栏。添加 `.uk-nav-offcanvas` 类来根据抽屉的上下文定义导航菜单的样式。

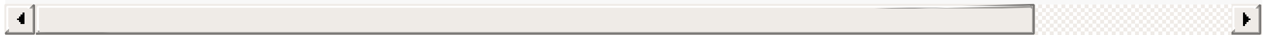
示例



Code

```
<!-- 这是用来触发抽屉式边栏的按钮 -->
<button class="uk-button" data-uk-offcanvas="{target:'#my-id'}">...

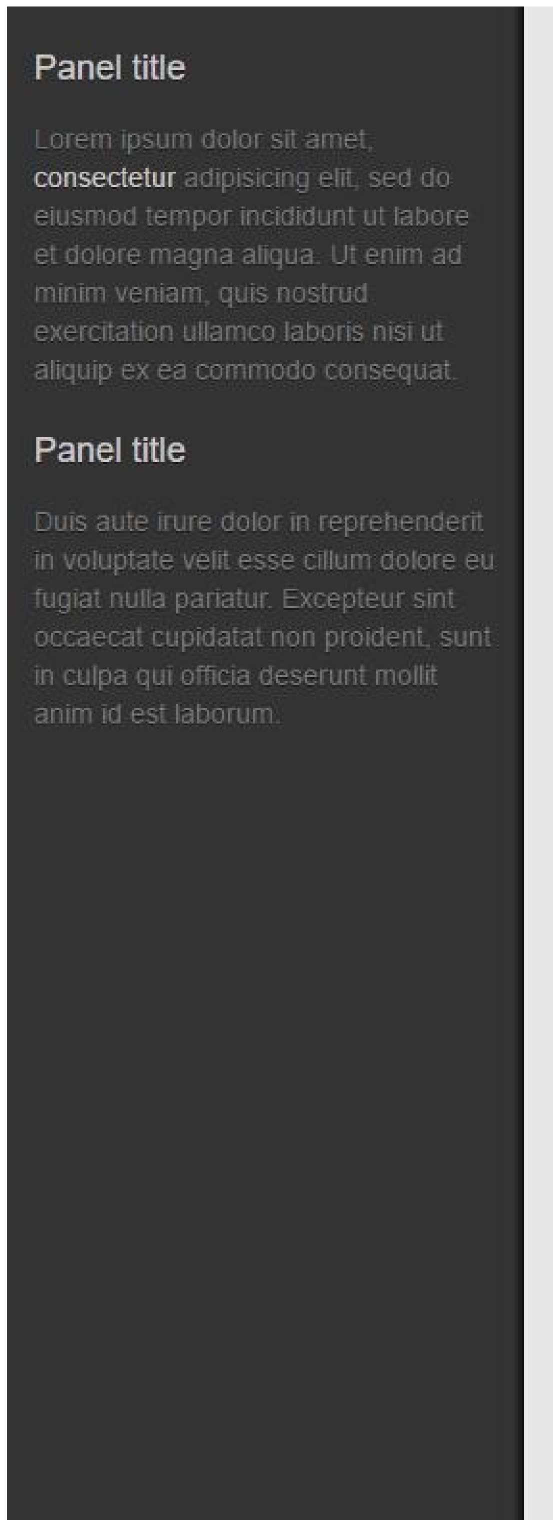
<!-- 抽屉式边栏 -->
<div id="my-id" class="uk-offcanvas">
    <div class="uk-offcanvas-bar">
        <ul class="uk-nav uk-nav-offcanvas" data-uk-nav>...</ul>
    </div>
</div>
```



抽屉式面板

你可以在抽屉式边栏的内部放置任何自定义内容。仅需将其包裹在带有 `.uk-panel` 类的 `<div>` 元素中即可。

示例


Panel title
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.
Panel title
Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Code

```
<!-- 这是用来触发抽屉式边栏的按钮 -->
<button class="uk-button" data-uk-offcanvas="{target:'#my-id'}">..

<!-- 这是抽屉式边栏 -->
<div id="my-id" class="uk-offcanvas">
    <div class="uk-offcanvas-bar">
        <div class="uk-panel">...</div>
    </div>
</div>
```

JavaScript

可以通过JavaScript来打开或关闭抽屉：

```
// 通过CSS选择器匹配抽屉显示状态
$.UIkit.offcanvas.show('#my-id');

// 隐藏当前所有抽屉。如果你不需要任何动画效果，设置为 true。
$.UIkit.offcanvas.hide([force = false])
```

事件

事件名称	参数	描述
`show.uk.offcanvas`	event, panel, bar	抽屉显示时触发
`hide.uk.offcanvas`	event, panel, bar	抽屉隐藏时触发

切换器/Switcher

通过不同的内容窗格进行动态变换。

用法

切换器组件由若干拨动器和与它们相关联的内容项目组成。添加 `data-uk-switcher="{connect:'#ID'}"` 到包含拨动器的元素，将此属性内的ID指向具有相同ID的包含内容项目的元素。添加 `.uk-switcher` 类到同一个元素中。通常，切换器和其他组件搭配使用，这里将展示其中一部分。

示例



Code

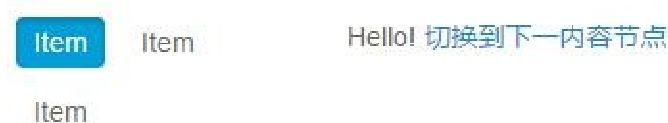
```
<!-- 这是包含拨动元素的容器 -->
<ul data-uk-switcher="{connect:'#my-id'}">
    <li><a href="">...</a></li>
</ul>

<!-- 这是内容项目的容器 -->
<ul id="my-id" class="uk-switcher">
    <li>...</li>
</ul>
```

在内容中切换条目/Switch items from within content

在某些情况下，你想要从显示的内容中操作切换到另一个内容节点。添加 `data-uk-switcher-item` 属性使之成为可能。为了指向目标条目，你必须将 data 属性设置为相应的标号。

Example



Markup

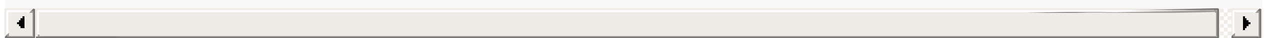
```
<!-- 这是包含拨动元素的导航 -->
<ul data-uk-switcher="{connect:'#my-id'}">
    <li><a href="">...</a></li>
</ul>

<!-- 这是内容条目的容器 -->
<ul id="my-id" class="uk-switcher">
    <li> ... <a href="" data-uk-switcher-item="0">...</a></li>
    <li> ... <a href="" data-uk-switcher-item="1">...</a></li>
</ul>
```

将 data 属性设置为 `next` 和 `previous` 将会切换到相邻的条目

Markup

```
<li> ... <a href="" data-uk-switcher-item="next">...</a></li>
<li> ... <a href="" data-uk-switcher-item="previous">...</a></li>
```



关联多个项目

切换器也可以管理多个内容容器。只需要在 `connect` 参数中填入需要关联的容器的ID，用逗号隔开。

示例



Code

```
<!-- 这是包含拨动元素的导航菜单 -->
<ul data-uk-switcher="{connect:'#my-id-one, #my-id-two'}">
    <li><a href="">...</a></li>
</ul>

<!-- 这些是包含着内容项目的容器们 -->
<ul id="my-id-one" class="uk-switcher">
    <li>...</li>
</ul>

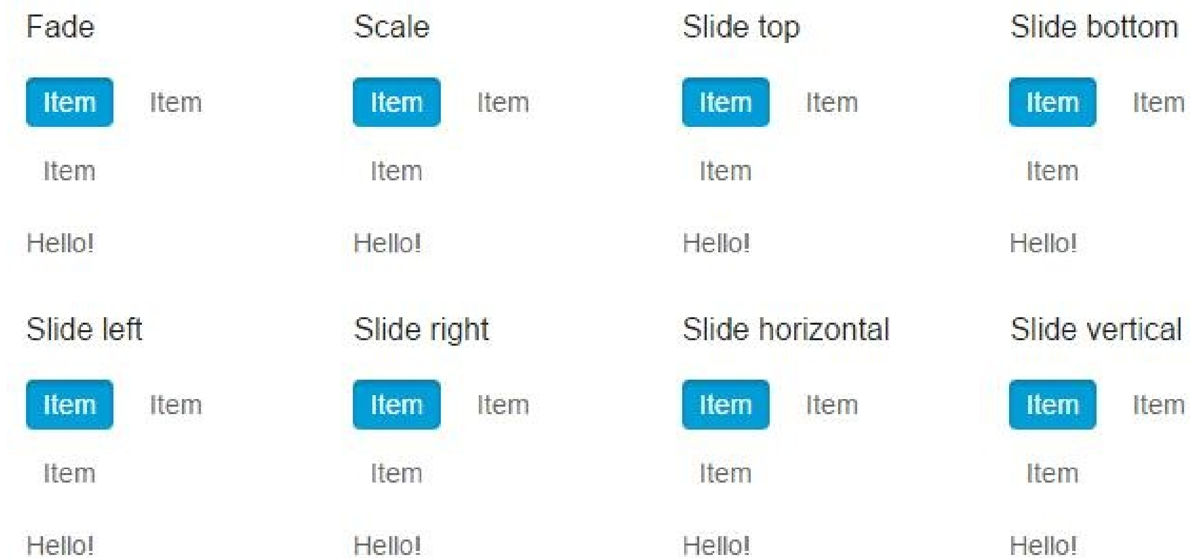
<ul id="my-id-two" class="uk-switcher">
    <li>...</li>
</ul>
```

动画

切换器组件允许你为它添加各式的切换动画。你所需要做的就是添加 `animation` 参数到data属性中，然后设置你希望使用的动画。查阅下面的列表，了解我们提供的动画吧。

Class	描述
`fade`	元素淡入
`scale`	元素由小变大
`slide-top`	元素从顶部滑入
`slide-bottom`	元素从底部滑入
`slide-left`	元素从左边滑入
`slide-right`	元素从右边滑入
`slide-horizontal`	内容项目水平方向滑动，方向基于元素相邻关系。
`slide-vertical`	内容项目垂直方向滑动，方向基于元素相邻关系。

示例



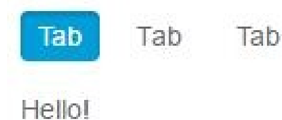

```
<!-- 包含拨动元素的容器 -->
<ul data-uk-switcher="{connect:'#my-id', animation: 'fade'}">
    <li><a href="">...</a></li>
</ul>

<!-- 包含内容项目的容器 -->
<ul id="my-id" class="uk-switcher">
    <li>...</li>
</ul>
```

自定义动画

你同样可以使用 动画组件 中的 `uk-animation-*` 类来应用多个动画。通过这种方式你甚至可以创建你的自定义类来为切换器应用不同的动画效果。

示例



Code

```
<!-- 包含拨动元素的容器  -->
<ul data-uk-switcher="{connect:'#my-id', animation: 'uk-animation-f
    <li><a href="">...</a></li>
</ul>

<!-- 包含内容项目的容器 -->
<ul id="my-id" class="uk-switcher">
    <li>...</li>
</ul>
```

切换器与二级导航

轻松地将切换器应用于 二级导航组件 中.

示例



Code

```
<!-- 这是包含拨动元素的二级导航 -->
<ul class="uk-subnav uk-subnav-pill" data-uk-switcher="{connect:'#m
    <li><a href="">...</a></li>
</ul>

<!-- 包含内容项目的容器 -->
<ul id="my-id" class="uk-switcher">
    <li>...</li>
</ul>
```

切换器与选项卡

作为一个特例，添加 `data-uk-tab="{connect:'#ID'}"` 属性，使用“tab”参数替代“switcher”，使得切换器和 选项卡组件 相互结合。

示例



Code

```
<!-- 这是包含拨动元素的选项卡式导航 -->
<ul class="uk-tab" data-uk-tab="{connect:'#my-id'}">
    <li><a href="">...</a></li>
</ul>

<!-- 包含内容项目的容器 -->
<ul id="my-id" class="uk-switcher uk-margin">
    <li>...</li>
</ul>
```

注意 当使用选项卡组件的 `.uk-tab-bottom` 类的时候，导航和内容需要按相反的顺序放置，这样内容才会显示在选项卡的上方。

垂直选项卡

使用 网格组件正确地显示带有垂直排列的选项卡导航的内容。

示例



Code

```
<!-- 位于左边的选项卡式导航 -->
<div class="uk-grid">
    <div class="uk-width-medium-1-2">
        <!-- 包含拨动元素的垂直选项卡式导航 -->
        <ul class="uk-tab uk-tab-left" data-uk-tab="{connect:'#my-i
    </div>
    <div class="uk-width-medium-1-2">
        <!-- 包含内容项目的容器 -->
        <ul id="my-id" class="uk-switcher">...</ul>
    </div>
</div>

<!-- 位于右边的选项卡式导航 -->
<div class="uk-grid">
    <div class="uk-width-medium-1-2 uk-push-1-2">
        <!-- 包含拨动元素的垂直选项卡式导航 -->
        <ul class="uk-tab uk-tab-right" data-uk-tab="{connect:'#my-
    </div>
    <div class="uk-width-medium-1-2 uk-pull-1-2">
        <!-- 包含内容项目的容器 -->
        <ul id="my-id" class="uk-switcher">...</ul>
    </div>
</div>
```

切换器与按钮

切换器同样可以应用于 按钮或按钮组 中。只需要添加切换器data属性到包含了一组按钮的 `<div>` 元素中或者添加到带有 `.button-group` 类的元素。

示例



Code

```
<!-- 包含拨动按钮的容器 -->
<div data-uk-switcher="{connect:'#my-id'}">
    <button class="uk-button" type="button">...</button>
</div>

<!-- 包含内容项目的容器 -->
<ul id="my-id" class="uk-switcher uk-margin">...</ul>

<!-- 包含拨动按钮的按钮租 -->
<div class="uk-button-group" data-uk-switcher="{connect:'#my-id'}">
    <button class="uk-button" type="button">...</button>
</div>

<!-- 包含内容项目的容器 -->
<ul id="my-id" class="uk-switcher uk-margin">...</ul>
```

切换器与导航菜单

若需将切换器应用于 导航菜单组件，添加data属性到导航菜单的 `<ul>` 元素中。使用 网格组件 将导航菜单和内容项放置在同一个网格布局中。

示例



Code

```
<div class="uk-grid">
    <div class="uk-width-medium-1-4">

        <!-- 包含拨动元素的导航菜单 -->
        <ul class="uk-nav uk-nav-side" data-uk-switcher="{connect:
            <li><a href="">...</a></li>
        </ul>
    </div>
    <div class="uk-width-medium-3-4">

        <!-- 包含内容项目的容器 -->
        <ul id="my-id" class="uk-switcher">
            <li>...</li>
        </ul>
    </div>
</div>
```

JavaScript 选项

这是一个关于如何通过属性设置选项的例子：

```
data-uk-switcher="{active:1}"
```

选项	可能的值	默认值	描述
`connect`	CSS selector	false	被关联的条目容器
`toggle`	CSS selector	'> *'	拨动CSS选择器，通过点击触发内容的切换行为。
`active`	integer	0	初始化时，呈激活状态的内容项目索引值
`animation`	mixed	false	预定义的动画名称或 uikit动画的类名。
`swiping`	boolean	true	启用或禁用通过滑动改变内容

事件

你可以为以下事件绑定回调函数，用于自定义功能：

事件名称	参数	描述
`show.uk.switcher`	event, active item	切换器条目显示或改变时触发

Example

```
$('[data-uk-switcher]').on('show.uk.switcher', function(event, area
    console.log("Switcher switched to ", area);
});
```

拨动/toggle

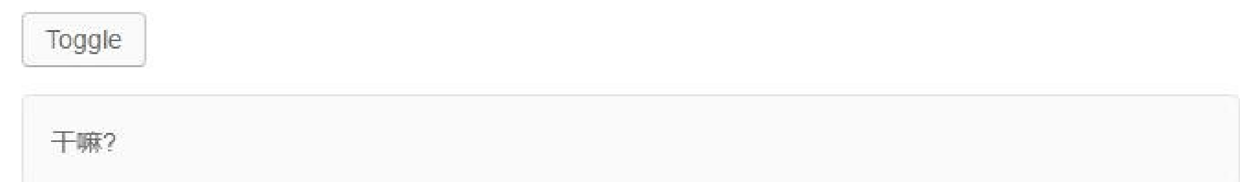
使用一个拨动器来隐藏，切换，或者改变各种内容的外观。

用法

要使用这个组件，只需要添加 `data-uk-toggle="{target: #ID}"` 属性到一个 `<button>` 或 `<a>` 元素中。你可以在任意选择器中使用这个拨动属性。

拨动器会在页面的某元素中添加或者删除一个class类。默认情况下，它会添加 `.uk-hidden` 类来隐藏这个元素。

示例



干嘛?

Code

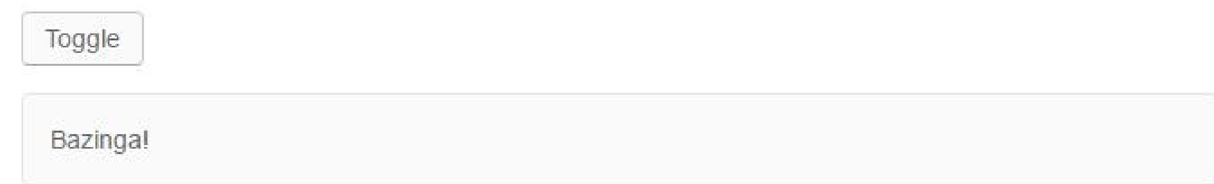
```
<button class="uk-button" data-uk-toggle="{target:'#my-id'}">...</

<div id="my-id">...</div>
```

拨动多个元素

你也可以同时拨动多个元素。使用适当的选择器就行。

示例



注意 在这个例子中，我们为其中一个元素添加了 `.uk-hidden` 类，所以只有另一个元素会显示出来。拨动器会在两个元素之间切换可见状态。

Code

```
<button class="uk-button" data-uk-toggle="{target:'.my-class'}">..

<div class="my-class">...</div>
<div class="my-class uk-hidden">...</div>
```

自定义class类

如果你不想切换 `.uk-hidden` 类，你也可以添加你的自定义类。

示例



注意 在这个例子中我们使用了 `.uk-panel-box-primary` 类来切换不同的面板样式。

Code

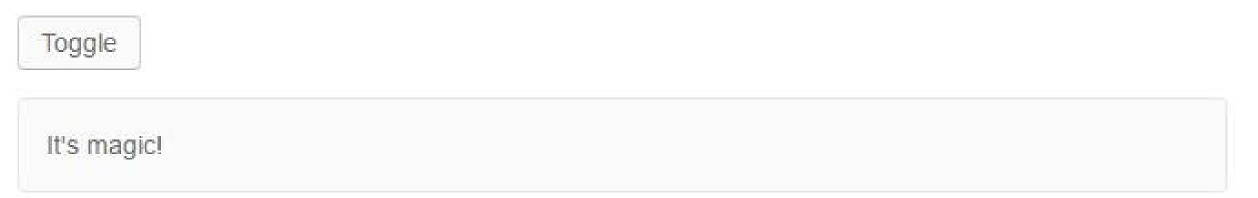
```
<button class="uk-button" data-uk-toggle="{target:'#my-id', cls:'uk

<div id="my-id" class="uk-panel uk-panel-box">...</div>
```

动画

拨动器组件允许你为元素添加拨动的动画效果。只需要添加一个 动画组件 中的 `.uk-animation-*` 类到 `animation` 参数中。 元素出现和消失都使用这个类。如果你喜欢不同的动画，只需添加另一个动画类就行了。

示例



Code

```
<button class="uk-button" data-uk-toggle="{target:'#my-id', animat:

<div id="my-id">...</div>
```

JavaScript 选项

这是一个例子，关于如何通过属性来设置选项:

```
data-uk-toggle="{target:'#my-id'}"
```

选项	可能的值	默认值	描述
`target`	CSS selector	false	被拨动的是哪个元素
`cls`	string	'uk-hidden'	拨动的class类
`animation`	mixed	false	任意 uikit 的动画类名

滚动监听/Scrollspy

在滚动页面时，触发一些事件及动画。

用法

滚动监听组件监听页面滚动，并触发基于滚动位置的事件。例如，当你向下滚动页面时，你可以使首次出现在视窗中的一个元素触发一个平滑淡入的动画。只需添加带有以下选项的 `data-uk-scrollspy` 属性。

Data 属性	描述
`cls:'MY-CLASS'`	只有元素首次出现在视窗时才应用这个属性中的class。
`repeat: true`	元素每次出现在视窗中时，都应用这个类。
`delay:600`	添加以毫秒为单位的动画延迟。

通常，动画组件中的类与滚动监视一起搭配使用。

示例

下面的例字使用了 `repeat: true` 选项。向上或向下滚动可以看到被触发的动画效果。

Fade

Lorem ipsum dolor sit amet, consectetur adipisicing elit.

Scale up

Lorem ipsum dolor sit amet, consectetur adipisicing elit.

Scale down

Lorem ipsum dolor sit amet, consectetur adipisicing elit.

Slide top

Lorem ipsum dolor sit amet, consectetur adipisicing elit.

Slide bottom

Lorem ipsum dolor sit amet, consectetur adipisicing elit.

Markup



```
<div data-uk-scrollspy="{cls:'uk-animation-fade'}">...</div>

<div data-uk-scrollspy="{cls:'uk-animation-fade', repeat: true}">.

<div data-uk-scrollspy="{cls:'uk-animation-fade', delay:900}">...<
```

组

你还可以将多个需要添加滚动监听效果的元素编成一组，这样就不必分别为每个元素添加 data 属性了。只需要添加 `data-uk-scollspy"{target:'MY-CLASS'}"` 属性到容器元素，将 target 选项指向容器中你想要添加动画效果的条目。当使用 delay(延时)时，将会为进入视野的一行条目添加逐次显现的效果。这个延时效果会为同一组内的下一行元素滚动进入视野时重置。

Example



Example



```
<div data-uk-scrollspy="{cls:'uk-animation-fade', target:'.my-class

    <!-- 无延时 -->
    <div class="my-class">...</div>

    <!-- 300ms 延时 -->
    <div class="my-class">...</div>

    <!-- 600ms 延时 -->
    <div class="my-class">...</div>

</div>
```

JavaScript选项

这是一个如何通过属性设置选项的示例：

```
data-uk-scrollspy="{cls:'uk-animation-fade'}"
```

选项	可用的值	默认值	描述
`cls`	string	'uk-scrollspy-inview'	当元素出现在视口内时添加的类。
`initcls`	string	'uk-scrollspy-init-inview'	当元素首次出现在视口内时添加的类。
`topoffset`	integer	0	在视口中触发事件前的顶部偏移量。
`leftoffset`	integer	0	在视口中触发事件前的左部偏移量。
`repeat`	boolean	false	元素是否每次出现在视口中都应用 `cls` 中提到的类。
`delay`	integer	0	以毫秒为单位的延时。

事件

你可以为以下事件绑定回调函数，以实现自定义功能：

名称	描述
`uk.scrollspy.init`	当元素开始进入视口时触发。
`uk.scrollspy.inview`	当元素在视口中时触发。
`uk.scrollspy.outview`	当元素离开视口时触发。

示例

```
$('[data-uk-scrollspy]').on('uk.scrollspy.inview', function(){
    // custom code...
});
```

滚动监听导航/Scrollspy Nav

根据你网站滚动条的位置来自动更新所对应的导航状态，只需添加Data属性 `data-uk-scrollspy-nav` 到任意导航栏中。每个菜单项必须链接至网站中网站中对应的具有相同ID的部分。

Data属性	描述
`data-uk-scrollspy-nav`	触发滚动监听导航的功能必须的JavaScript。
`data-uk-scrollspy-nav="{closest:'MY-SELECTOR'}"`	通过遍历查找此DOM树中最接近的节点元素进行匹配。
`data-uk-scrollspy-nav="{smoothscroll:true}"`	在网页的不同部分之间跳转时，应用平滑滚动组件。
`data-uk-scrollspy-nav="{cls:'MY-CLASS'}"`	默认地，滚动监听导航会拨动 `uk-active` 类。使用 `cls` 选项使用你自己的类名。

对于滚动监听的例子，可以查看滚动监听测试页面。

Code

```
<ul class="uk-nav uk-nav-side" data-uk-scrollspy-nav="{closest:'li
    <li><a href="#MY-ID">...</a></li>
    <li><a href="#MY-ID2">...</a></li>
</ul>

<div id="MY-ID">...</div>
<div id="MY-ID2">...</div>
```

JavaScript选项

这是一个如何通过属性设置选项的示例：

```
data-uk-scrollspy-nav="{smoothscroll:true}"
```

选项	可用的值	默认值	描述
`cls`	string	'uk-active'	添加其中的class以激活元素。
`closest`	CSS 选择器	false	将上面所述的类，应用到本选择器对应的元素中。
`topoffset`	integer	0	滚动的顶部偏移量。
`leftoffset`	integer	0	滚动的左部偏移量。
`smoothscroll`	boolean	false	是否显示平滑的滚动动画效果。

事件

名称	参数	描述
`init.uk.scrollspy`	event	滚动监听第一次初始化时触发
`inview.uk.scrollspy`	event	元素进入视口后触发
`outview.uk.scrollspy`	event	元素离开视口后触发

平滑滚动

用一个漂亮的缓慢滚动效果来增强网站页面内各部分之间跳转时的效果。

用法

为了能在网页中的一部分跳转到另一部分时，渐渐降低跳转的速度，例如，一个回到顶部的按钮，只需要添加 `data-uk-smooth-scroll` 属性到一个链接到目标位置的 `<a>` 元素中即可。

示例

在我们的例子中，使用了标题的ID作为 平滑滚动 的目标。



Code

```
<a href="#my-id" class="uk-button" data-uk-smooth-scroll>...</a>
```

偏移

在精确计算滚动位置时，通过偏移选项来添加相对于目标的指定距离。偏移量通过data属性传递。它的数值为正时，在到达目标位置之前停止滚动；数值为负时，在超过目标位置后停止滚动。

Data 属性	描述
`data-uk-smooth-scroll="{offset: 90}"`	添加一个偏移量，在滚动到目标还有90PX的位置停下来。
`data-uk-smooth-scroll="{offset: -90}"`	添加一个偏移量，在滚动到超过目标位置90PX的地方停下来。

示例

This link scrolls the site to the headline "Smooth scroll" with an offset of `90px`.

This link scrolls the site to the headline "Smooth scroll" with an offset of `-90px`.

Code

```
<a href="#my-id" data-uk-smooth-scroll="{offset: 90}">...</a>

<a href="#my-id" data-uk-smooth-scroll="{offset: -90}">...</a>
```

附加组件

> 来源：附加组件

各种有用的独立附加组件，它们并不包含在 UIKit 的核心框架中。

UIKit 提供了一些高级的组件，它们不包含在 UIKit 的核心框架中。通常你在普通的网站上不会用到这些插件。如果你创建高级的用户界面管理等领域，它们便会派上用场，比如“拖放排序”和“日期选择器”。

用法

当你 后，你可以在 `/css` 和 `/js` 文件夹里的 `/components` 中发现附加组件的相关文件:

```
/css
    /components
        <!-- 组件的基本样式 -->
        autocomplete.css
        autocomplete.min.css

        <!-- 组件具有渐变风格的样式 -->
        autocomplete.gradient.css
        autocomplete.gradient.min.css

        <!-- 组件具有扁平化风格的样式 -->
        autocomplete.almost-flat.css
        autocomplete.almost-flat.min.css
        ...

/js
    /components
        <!-- 组件的JavaScript和压缩后的版本 -->
        autocomplete.js
        autocomplete.min.js
        ...
```

注意 若要使用这些附加组件，你必须在文档头部添加它的JavaScript和CSS文件。

Example

```
<!DOCTYPE html>
<html>
    <head>
        <title></title>
        <link rel="stylesheet" href="uikit.min.css" />
        <script src="jquery.js"></script>
        <script src="uikit.min.js"></script>
        <!-- Autocomplete CSS -->
        <link rel="stylesheet" href="components/autocomplete.css">
        <!-- Autocomplete Javascript -->
        <script src="components/autocomplete.js"></script>
    </head>
    <body>
    </body>
</html>
```

Tests

UIkit为每一个组件提供了测试文件。每个测试页面包含其组件的测试案例和开箱即用的所有可能性。

在ZIP压缩包里你可以找到UIkit为你的项目提供的所有CSS，JavaScript和字体文件。为了保持UIkit的核心框架的简洁，核心框架中几乎不包含任何样式。因此我们提供两套额外的样式——包括渐变风格和扁平风格。每一种样式都包含一个单独的CSS文件和一个压缩版的CSS文件。一些并不包含在核心框架中的附加组件也提供了额外的两套CSS文件，它们可以被单独的引入到你的项目中使用。

前往测试

布局类组件

动态网格

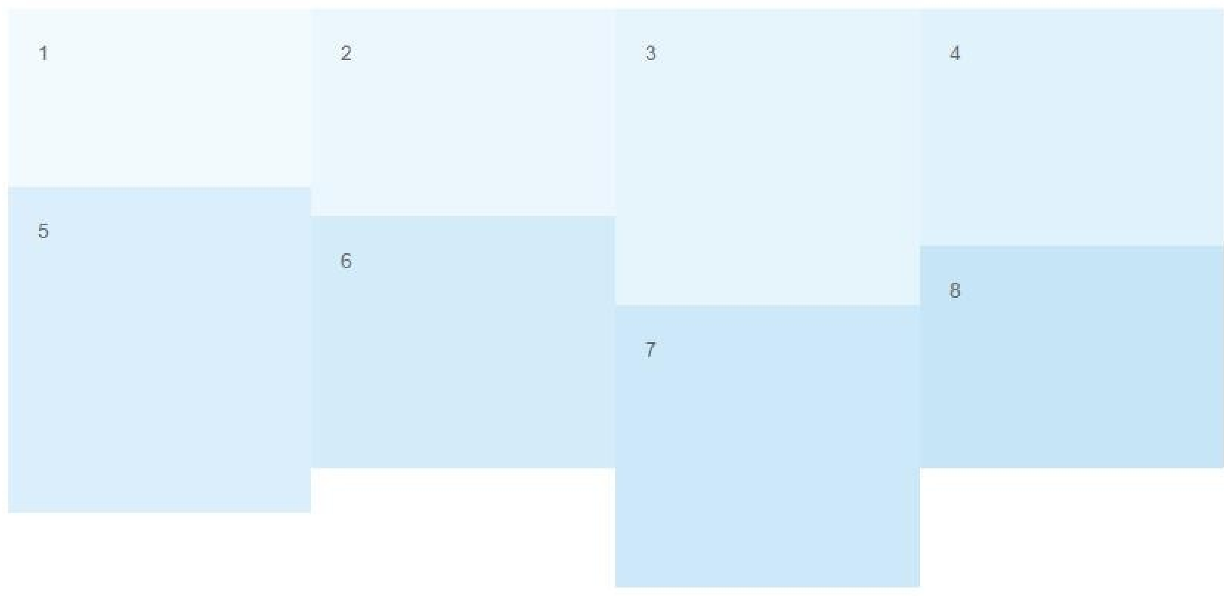
创建可排序、可过滤的多列动态网格布局。

动态网格组件允许你利用网格组件创建一个动态的响应式网格布局。在任何尺寸的设备中，网格单元都将自动地、流动地、无缝地调整自身构建无缝的多列布局。

用法

使用这个组件，添加 `data-uk-grid` 属性到容器元素。用 网格组件 中的 `uk-width-*` 或 `.uk-grid-width-*` 类来设置网格单元的宽度。注意 使用此组件需要额外添加 `grid.js` 文件，在 `js/components` 文件夹中。

Example



Markup

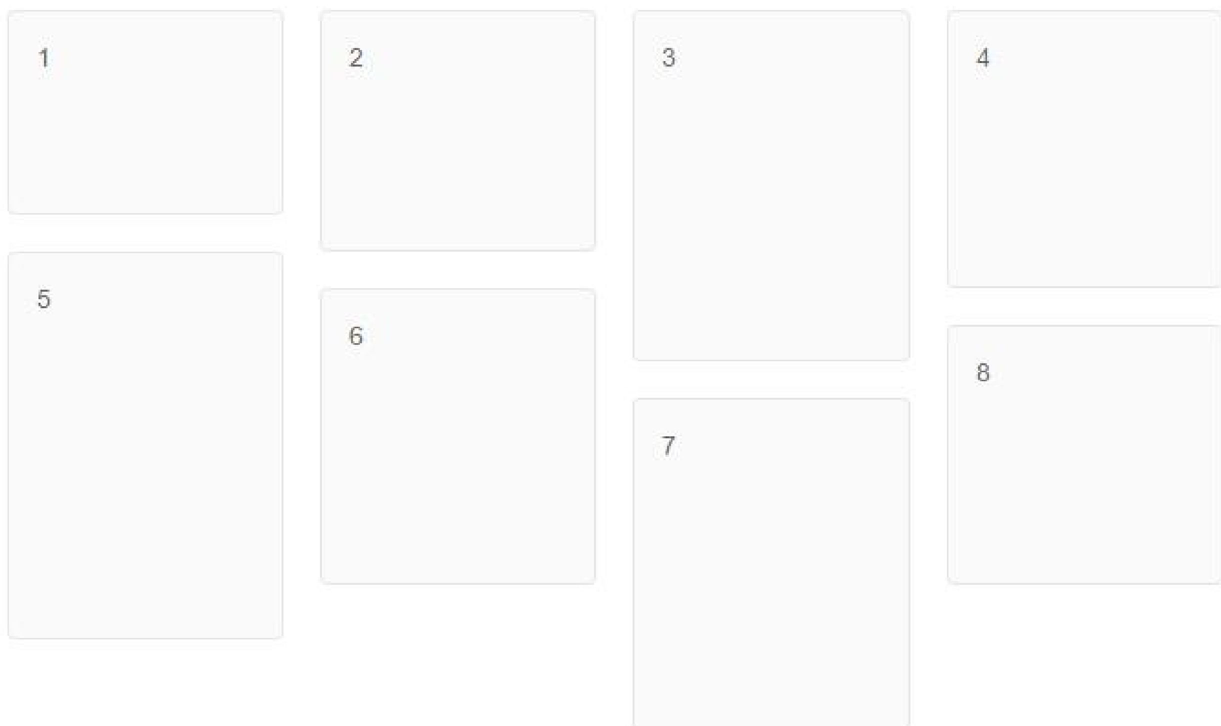
```
<!-- 这是使用在每个网格单元上使用 uk-width-* 的网格 -->
<div data-uk-grid>
    <div class="uk-width-small-1-2 uk-width-medium-1-4">...</div>
    <div class="uk-width-small-1-2 uk-width-medium-1-4">...</div>
</div>

<!-- 这是使用在网格自身使用 uk-grid-width-* 的网格 -->
<div class="uk-grid-width-small-1-2 uk-grid-width-medium-1-4" data-
    <div>...</div>
    <div>...</div>
</div>
```

网格排水沟/Grid Gutter

用排水沟将网格单元分开，在data属性中使用 `{gutter: 20}` 选项即可。此时，排水沟会是20px宽。

Example



Markup

```
<div data-uk-grid="{gutter: 20}">...</div>
```

过滤

你还可以过滤网格，使其只在页面中显示特定的网格单元。You can also filter your grid so that only particular items will be displayed. 为此，添加 `{controls: '#my-id'}` 选项到data属性指向过滤器的id。每个过滤控制器都必须有 `data-uk-filter` 属性来定义你想要的过滤分类。你还必须为每个网格单元添加 `data-uk-filter` 属性以明确其归属于哪个分类。使用逗号分隔多段分类。

Example



Markup

这个例子使用了 二级导航 来过滤网格单元。

```
<!-- Filter Controls -->
<ul id="my-id" class="uk-subnav">
    <li data-uk-filter=""><a href=""></a></li>
    <li data-uk-filter="filter-a"><a href=""></a></li>
    <li data-uk-filter="filter-b"><a href=""></a></li>
</ul>

<!-- Dynamic Grid -->
<div data-uk-grid="{controls: '#my-id'}">
    <div data-uk-filter="filter-a">...</div>
    <div data-uk-filter="filter-b">...</div>
    <div data-uk-filter="filter-a,filter-b">...</div>
</div>
```

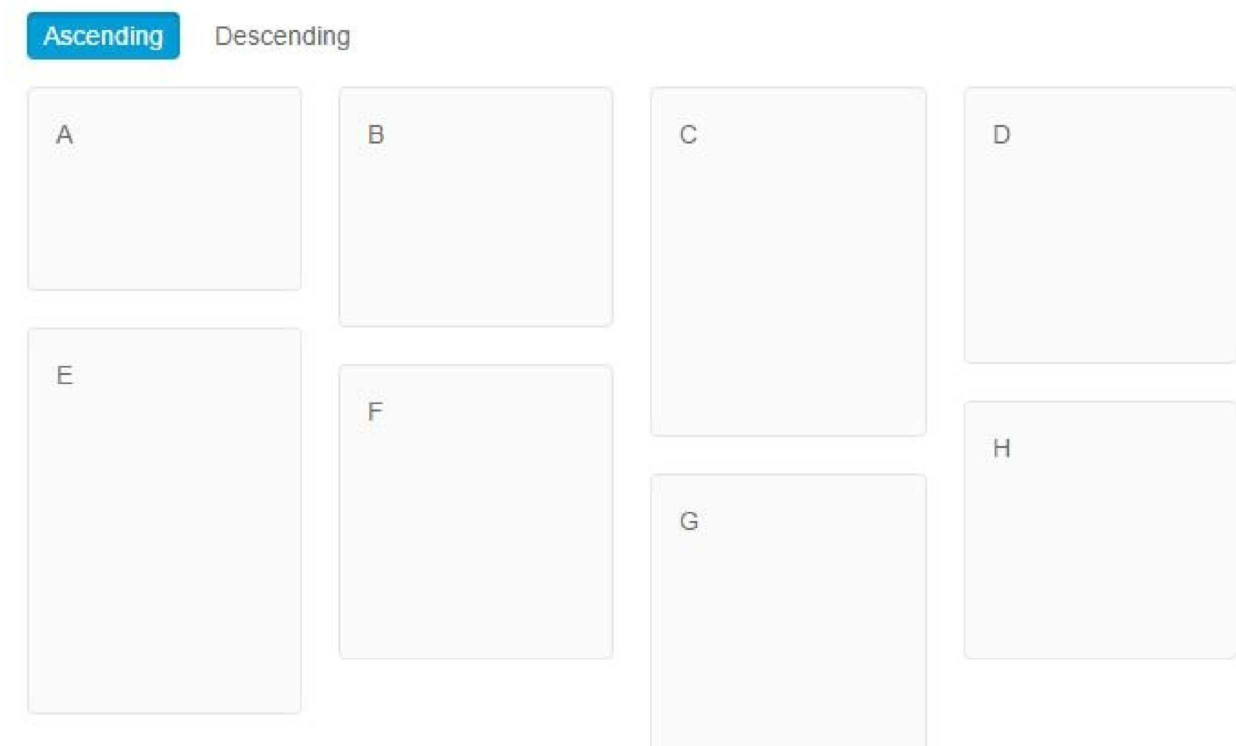
排序

升序排列网格单元，添加 `{controls: '#my-id'}` 选项到data属性指向排序控件的id。

每个控件需要带有 `data-uk-sort` 属性以及一个自定义值以指明你想要进行排序的种类，比如 `data-uk-sort="my-category"` 。同样，你需要添加带有指定分类名称的data属性到每个网格单元中。 自定义的值包含应该被排序的数据，比如 `data-my-category="my-data"` 。

网格单元默认按升序排序。要实现降序，只需添加 `:desc` 到排序控件的data属性值中，比如 `data-uk-sort="my-category:desc"` 。

Example



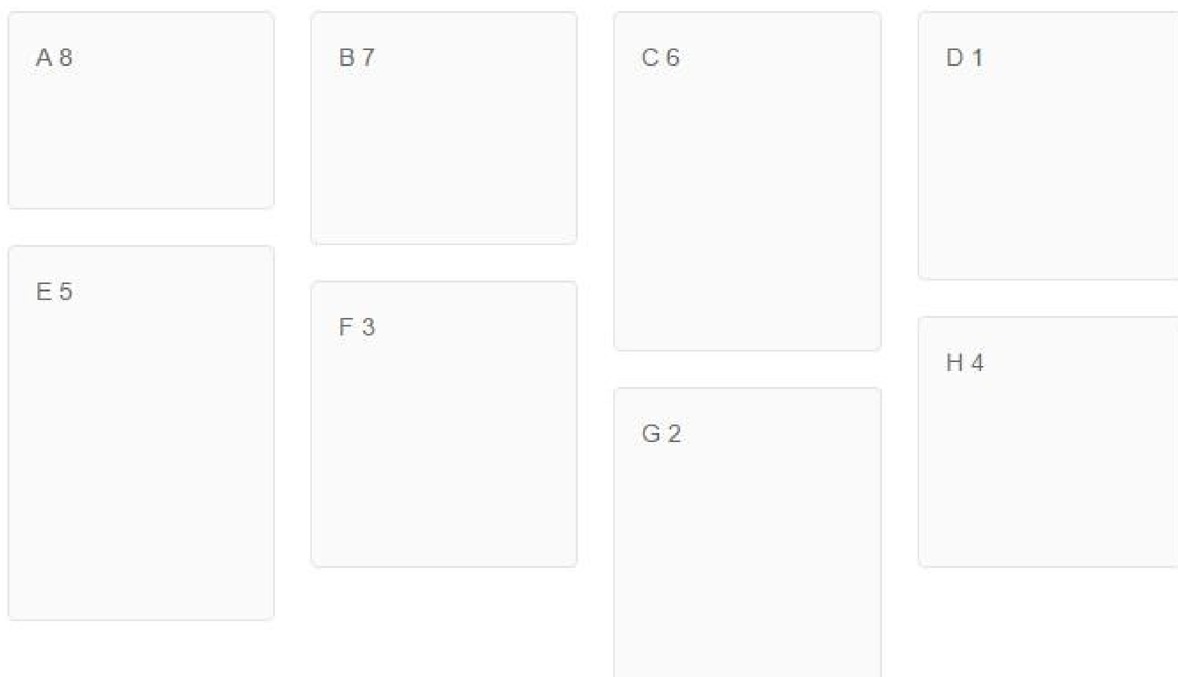
Markup

```
<ul id="my-id" class="uk-subnav">
    <li data-uk-sort="my-category"><a href=""></a></li>
    <li data-uk-sort="my-category:desc"><a href=""></a></li>
</ul>

<div data-uk-grid="{controls: '#my-id'}">
    <div data-my-category="a">...</div>
    <div data-my-category="b">...</div>
</div>
```

多个类别

要使用多个类别来进行网格单元的排序，为每个类别使用不同的名字就行了。



Markup

```
<ul id="my-id" class="uk-subnav">
    <li data-uk-sort="my-category"><a href=""></a></li>
    <li data-uk-sort="my-category:desc"><a href=""></a></li>
    <li data-uk-sort="my-category2"><a href=""></a></li>
    <li data-uk-sort="my-category2:desc"><a href=""></a></li>
</ul>

<div data-uk-grid="{controls: '#my-id'}">
    <div data-my-category="a" data-my-category2="8">...</div>
    <div data-my-category="b" data-my-category2="7">...</div>
</div>
```

JavaScript 选项

选项	可用值	默认值	描述
`colwidth`	integer	auto	列宽
`animation`	boolean	true	是否开启列刷新的动画
`duration`	integer	200	动画持续时间
`gutter`	integer	0	列与列之间的排水沟
`controls`	string	false	联结过滤器或排序控件的 CSS 选择器
`filter`	string	''	单元过滤器

手动初始化

```
var grid = UIkit.grid(element, { /* options */ });
```

Events

名称	参数	描述
`beforeupdate.uk.grid`	event, children	网格刷新前触发
`afterupdate.uk.grid`	event, children	网格刷新后触发

Example

用jQuery监听beforeupdate事件:

```
$(function() {
    $('[data-uk-grid]').on('beforeupdate.uk.grid', function(e, chi
        // your event-handling goes here
    });
});
```

视差网格

创建作用于单个网格列，使之具有不同滚动速度的效果。

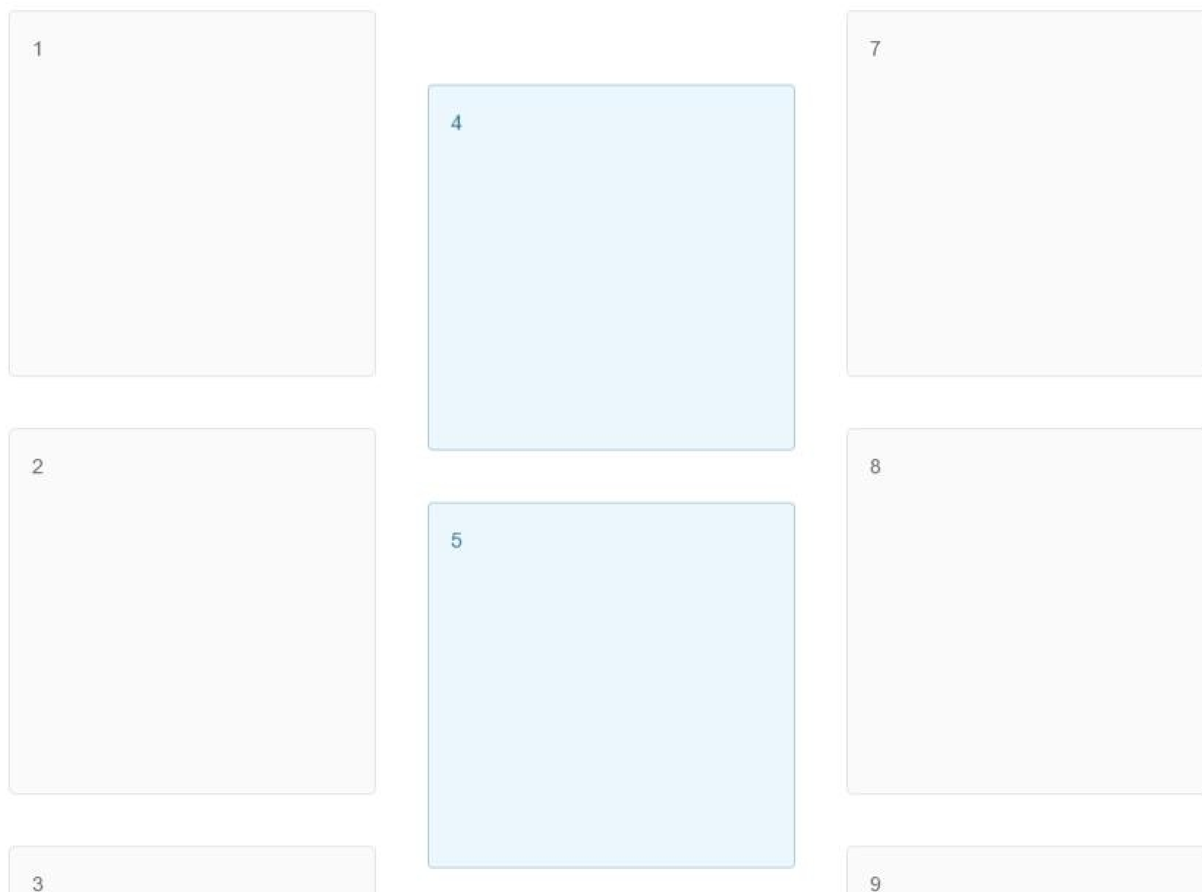
这个组件允许你为网格组件添加滚动视差效果。

用法

注意 使用此组件，必须引入 `grid-parallax.js` ,你可以在 `js/components/` 目录中找到它。

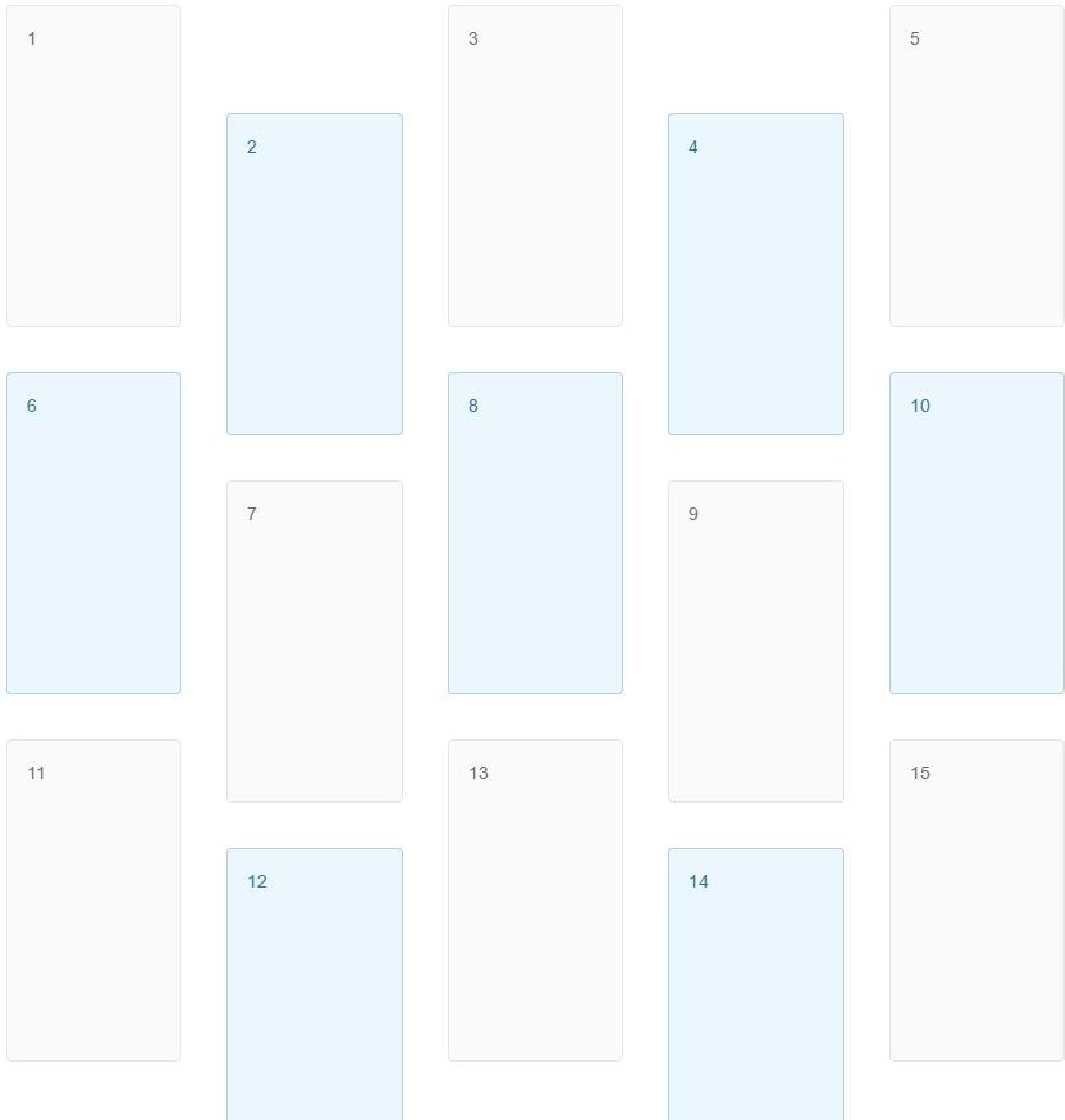
使用这个组件，需要为容器元素添加 `data-uk-gridparallax` 属性。使用网格组件中的 `uk-width-*` 或 `.uk-grid-width-*` 来设置网格条目单元的宽度。

Example



Javascript 参数

想要调整滚动速度，只需要改变 `data-uk-grid-parallax` 属性的参数即可，比如： `data-uk-grid-parallax="{translate:200}"`



JavaScript 选项

通过属性来设置选项的例子：

```
data-uk-grid-parallax="{translate:200}"
```

选项	可能的值	默认值	描述
`translate`	integer	150	Translate 数值

手动初始化

```
var gp = UIkit.gridparallax(element, { /* options */ });
```

导航类组件

圆点导航

创建水平或垂直布局的圆点导航，通过幻灯片或者滚动的方式导航到页面的不同部分。

用法

创建带有圆点的导航，只需要添加 `.uk-dotnav` 类到 `<ul>` 元素中，再将 `<a>` 元素嵌套进列表即可。于是，完美地创建咯一个典型的幻灯片导航。这个组件使用Flex进行构建，所以，你可以使用 Flex 调整圆点导航。注意 使用此组件需要额外添加 `dotnav.css` 文件，在 `css/components` 文件夹中。

Example



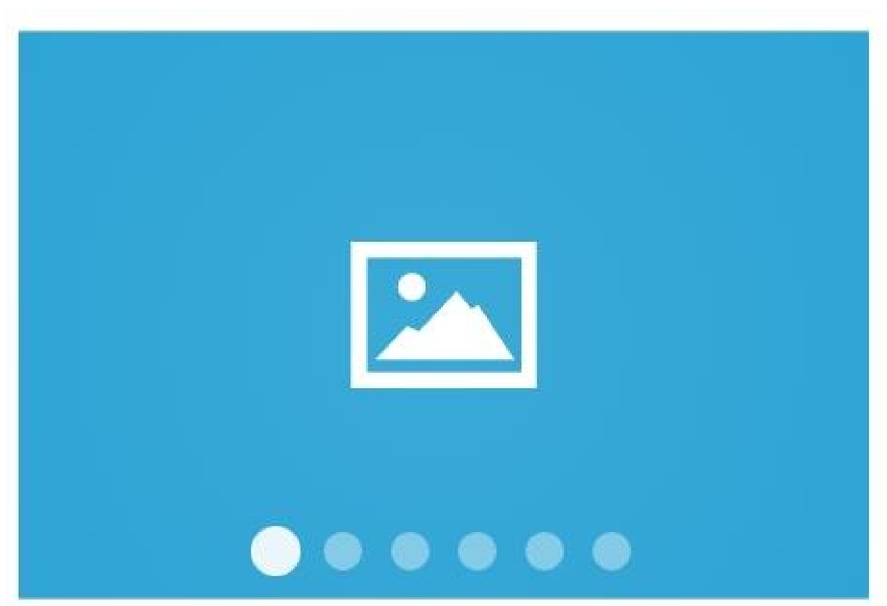
Markup

```
<ul class="uk-dotnav">
    <li class="uk-active"><a href="">...</a></li>
    <li><a href="">...</a></li>
</ul>
```

圆点导航与图片

为了能在图片上有更好的可见性，比如使用圆点导航作为幻灯片的导航时，只需要添加 `.uk-dotnav-contrast` 类。

Example



注意 在这个例子中，我们还用了Flex 组件 中的 `.uk-flex-center` 类来居中圆点导航。

Markup

```
<ul class="uk-dotnav uk-dotnav-contrast uk-flex-center">
    <li class="uk-active"><a href="">...</a></li>
    <li><a href="">...</a></li>
</ul>
```

垂直的圆点导航

圆点导航可以垂直对齐。只需添加 `.uk-flex-column` 类即可。这对于使用 滚动监听 进行页面滚动导航是非常有用的。

Example



Markup

```
<ul class="uk-dotnav uk-flex-column">
    <li class="uk-active"><a href="">...</a></li>
    <li><a href="">...</a></li>
</ul>
```

滑动导航/Slidenav

创建带有上一个和下一个按钮的导航，用来浏览幻灯片。

用法

创建有上一个和下一个按钮的导航，添加 `.uk-slidenav` 类到 `<a>` 元素就行。添加 `.uk-slidenav-previous` 或 `.uk-slidenav-next` 类来定义导航条目作为上一个和下一个按钮。注意 使用此组件需要额外添加 `slidenav.css` 文件，在 `css/components` 文件夹中。

Example

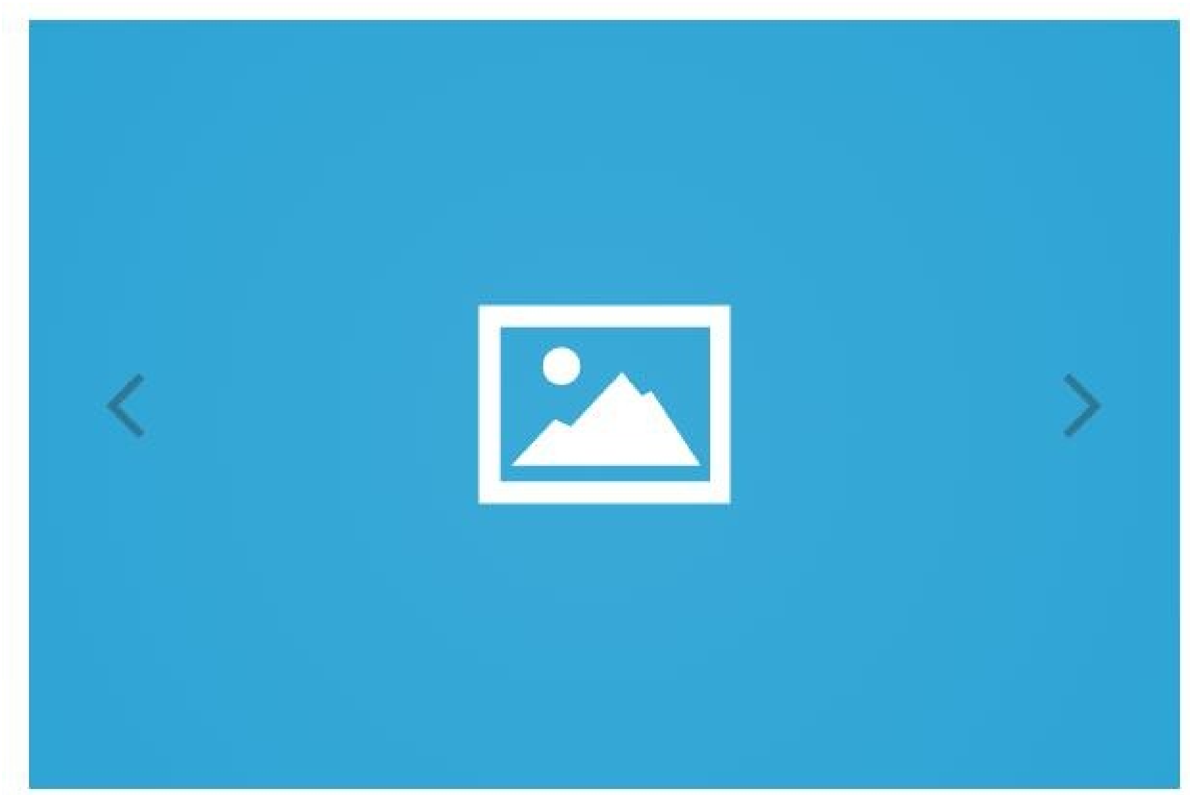


Markup

```
<a href="" class="uk-slidenav uk-slidenav-previous"></a>
<a href="" class="uk-slidenav uk-slidenav-next"></a>
```

滑动导航的定位

在任意内容上面定位滑动导航，比如在幻灯片或者图片上，只需要将导航有内容包含在一个容器元素中，并添加 `.uk-slidenav-position` 类即可。这样子，上一个和下一个按钮将会分别被垂直居中在左边和右边。滑动导航只在鼠标悬停在内容上时可见。

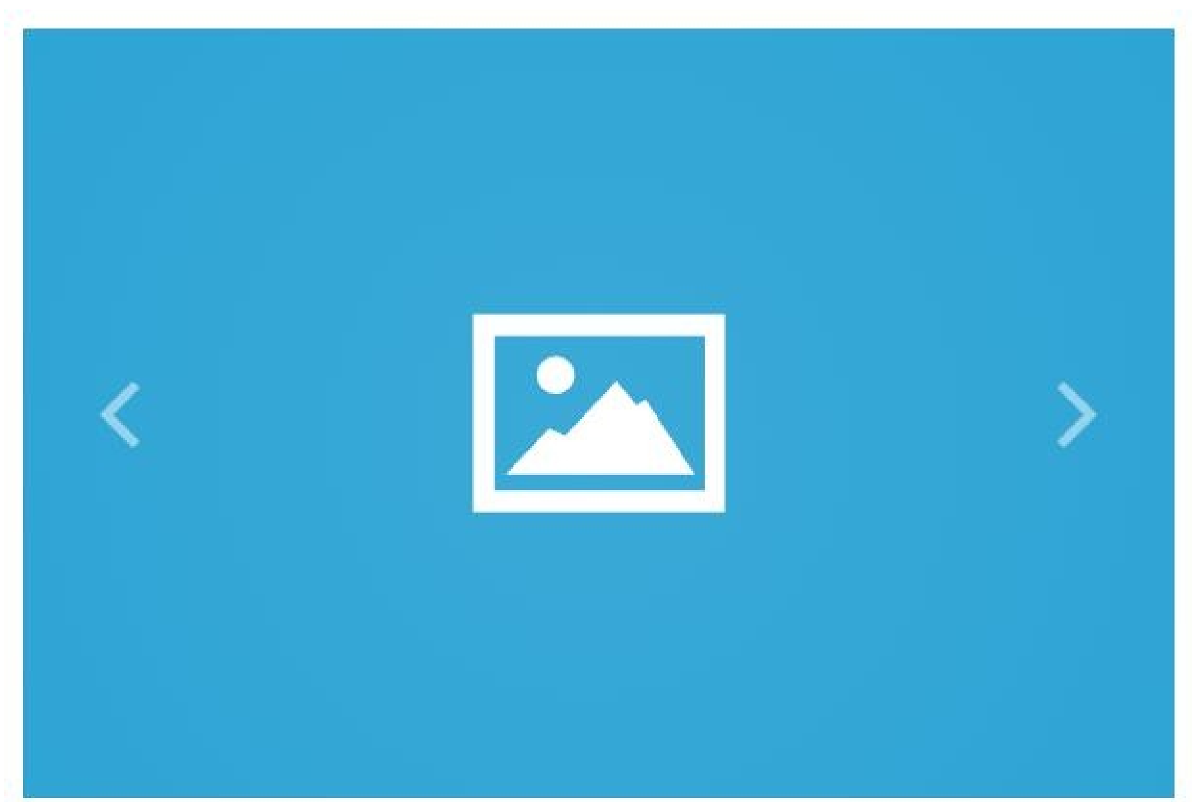


Markup

```
<div class="uk-slidenav-position">
    <img src="" alt="">
    <a href="" class="uk-slidenav uk-slidenav-previous"></a>
    <a href="" class="uk-slidenav uk-slidenav-next"></a>
</div>
```

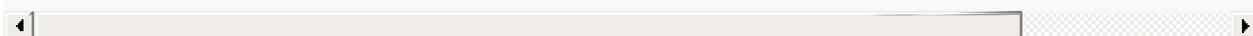
滑动导航与图片

为了在图片上有更好的可见性，比如使用滑动导航作为幻灯片的导航时，只需添加 `.uk-slidenav-contrast` 类。



Markup

```
<a href="" class="uk-slidenav uk-slidenav-contrast uk-slidenav-prev
<a href="" class="uk-slidenav uk-slidenav-contrast uk-slidenav-next
```



动态分页

利用分页组件，创建基于JavaScript的动态分页。

分页组件将根据给定的选项自动计算页面。这非常有用，例如在Ajax支持的列表视图中，你必须触发一个事件来动态地加载新的条目时。

用法

要使用这个组件，只需要添加 `data-uk-pagination` 到一个带有 `.uk-pagination` 类名的 `<ul>` 元素中。注意 使用此组件需要额外添加 `pagination.js` 文件，在 `js/components` 文件夹中。

Example



Markup

```
<ul class="uk-pagination" data-uk-pagination="{items:100, itemsOnPa
```

JavaScript 选项

选项	可用值	默认值	描述
`items`	integer	1	条目的总数，用于计算页面。
`itemsOnPage`	integer	1	每个页面上显示的条目数。
`pages`	integer	0	如果被指定了值, **items** and **itemsOnPage** 将不会用于计算页面数目。
`displayedPages`	integer	3	导航时有多少页码是可见的。
`edges`	integer	3	在分页的首页或末页有多少页面是可见的。
`currentPage`	integer	1	初始化后，哪一页应该被立即选中。

事件

名称	参数	描述
`select.uk.pagination`	event, pageIndex, pagination object	页面点击时

每当你点击某个页面时，分页组件便会触发 `uk-select-page` 事件。

```
$('[data-uk-pagination]').on('select.uk.pagination', function(e, pa
    alert('You have selected page: ' + (pageIndex+1));
});
```

手动初始化

```
var pagination = UIkit.pagination(element, { /* options */ });
```

常用组件

高级表单

为表单中的单选按钮和复选框添加自定义的样式。

UIkit 使用 外观 特性只使用CSS进行样式的定义，而非采用自定义标记。目前，外观特性只被webkit浏览器完整支持，所以这些样式只会在Chrome、Safari以及Opera浏览器上生效。一旦其他浏览器支持这个特性时，这个现状一定会改变的。

用法

你可以使用这个组件来扩展 表单组件。只需要讲 `<input>` 元素放入到 `<form>` 元素中并添加 `.uk-form` 类名。注意 使用此组件需要额外添加 `form-advanced.css` 文件，在 `css/components` 文件夹中。

Example

1 2 3

1 2 3

Markup

```
<form class="uk-form">
    <input type="radio">
    <input type="checkbox">
</form>
```

文件表单/Form file

用诸如按钮等自定义的HTML结构替代默认的文件域表单。

用法

若要应用此组件，只需添加 `.uk-form-file` 类到一个包含着按钮和 `<input type="file">` 元素的容器中。你也可以用其他任意元素替代按钮。注意 使用此组件需要额外添加 `form-file.css` 文件，在 `css/components` 文件夹中。

Example



Markup

```
<div class="uk-form-file">
    <button class="uk-button">...</button>
    <input type="file">
</div>

<div class="uk-form-file">Text<input type="file"></div>
```

密码表单

创建带有拨动密码显示或隐藏功能的密码表单。

用法

要使用这个组件，添加 `uk-form-password` 类到一个包含着密码输入框的 <div> 元素。创建拨动密码显示和隐藏的按钮，只需要添加 `.uk-form-password-toggle` 类和 `data-uk-form-password` 属性到 <a> 元素即可。注意 使用此组件需要额外添加 `form-password.css` 文件，在 `css/components` 文件夹中。此组件需要额外添加 `form-password.js` 文件，在 `js/components` 文件夹中。

Example

••••••••• Show

Markup

```
<form class="uk-form">
    <div class="uk-form-password">
        <input type="password">
        <a href="" class="uk-form-password-toggle" data-uk-form-pas
    </div>
</form>
```

JavaScript 选项

这是一个如何通过属性设置选项的例子：

```
data-uk-form-password="{lblShow:'...', lblHide:'...'}"
```

选项	可用值	默认值	描述
`lblShow`	任意 string	'Show'	隐藏标签文本
`lblHide`	任意 string	'Hide'	显示标签文本

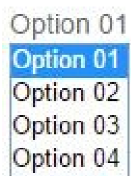
选择表单

使用按钮、文本等HTML内容替代默认的选择表单。

用法

使用这个组件，只需添加 `.uk-form-select` 类和 `data-uk-form-select` 属性到包含 `<span>` 和 `<select>` 元素的容器元素中。基本上，这将用自定义的UIkit表单覆盖默认的选择表单。注意 使用此组件需要额外添加 `form-select.css` 文件，在 `css/components` 文件夹中。此组件需要额外添加 `form-select.js` 文件，在 `js/components` 文件夹中。

Example

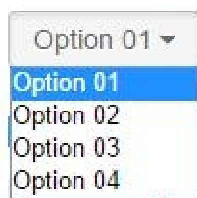


Markup

```
<div class="uk-form-select" data-uk-form-select>
    <span></span>
    <select>
        <option value="">...</option>
        <option value="">...</option>
    </select>
</div>
```

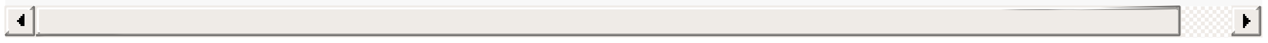
类似于按钮

添加 `.uk-button` 类到容器元素，使之看起来像个按钮。



Markup

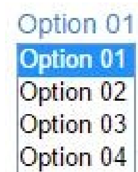
```
<div class="uk-button uk-form-select" data-uk-form-select>...</div>
```



类似于锚文本

你可以使用任意元素作为选择表单，比如 `<a>` 元素。只需添加 `{target:'a'}` 选项到 `data-uk-form-select` 属性，并使用 `<a>` 替代 `<span>` 元素。

Example



Markup

```
 <div class="uk-form-select" data-uk-form-select="{target:'a'}">
    <a></a>
    <select>...</select>
</div>
```

占位符

创建一个占位符空间，可以用于拖拽上传文件。

Usage

这个组件适用于设计一些特殊的区域来创建一个占位符空间. 只需要添加 `.uk-placeholder` 类到 <div> 元素中。注意 使用此组件需要额外添加 `placeholder.css` 文件，在 `css/components` 文件夹中。

Example



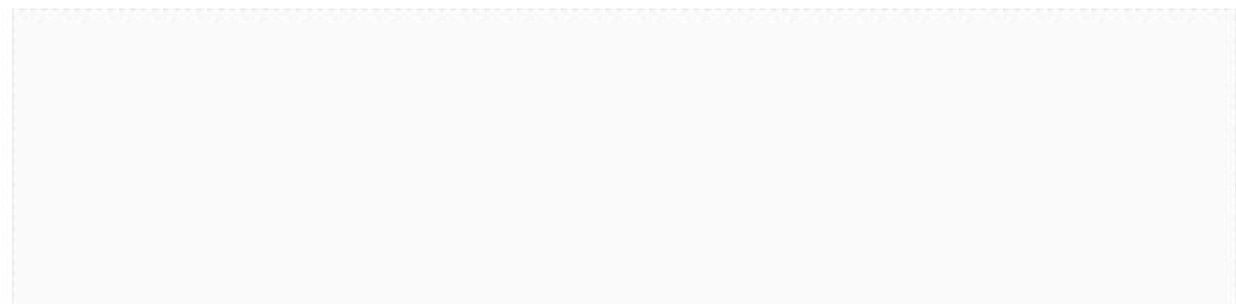
Markup

```
<div class="uk-placeholder">...</div>
```

大空间占位符

你还可以在占位符空间上增加高度，使用 `.uk-placeholder-large` 类。

Example



进度条/Progress

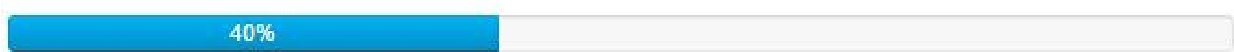
为进度条定义不同样式。

用法

进度条由背景和进度条本身组成，用来体现增长。注意 使用此组件需要额外添加 `progress.css` 文件，在 `css/components` 文件夹中。

Class	描述
`.uk-progress`	该类用在父元素来创建进度条的背景。
`.uk-progress-bar`	该类需要被添加至子元素用于创建实际的进度条。

Example



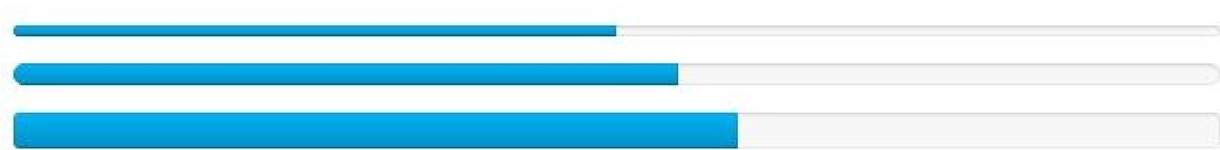
Markup

```
<div class="uk-progress">
    <div class="uk-progress-bar" style="width: 40%;">40%</div>
</div>
```

尺寸调整

添加 `.uk-progress-mini` 或 `.uk-progress-small` 类用于改变进度条的尺寸。

Example



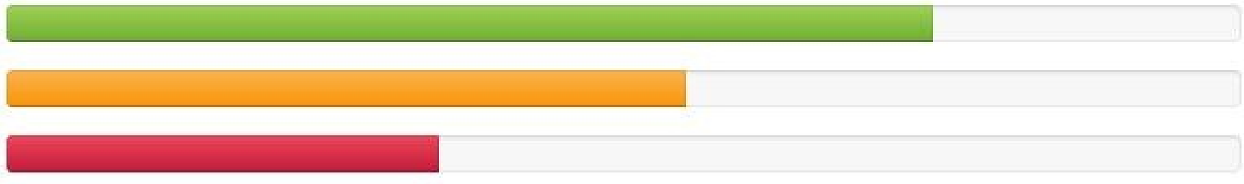
Markup

```
<div class="uk-progress uk-progress-mini">...</div>
<div class="uk-progress uk-progress-small">...</div>
```

颜色调整

要在您的进度条中应用不同的颜色，仅需添加 `.uk-progress-success` ，`.uk-progress-warning` 或 `.uk-progress-danger` 类。

Example



Markup

```
<div class="uk-progress uk-progress-success">...</div>
<div class="uk-progress uk-progress-warning">...</div>
<div class="uk-progress uk-progress-danger">...</div>
```

条纹

使用 `.striped` 类，创建一个带条纹的进度条。

Example



Markup

```
<div class="uk-progress uk-progress-striped">...</div>
```

甚至可以使条纹动起来。为此，仅需添加 `.uk-active` 类。

Example



Markup

```
<div class="uk-progress uk-progress-striped uk-active">...</div>
```

组合

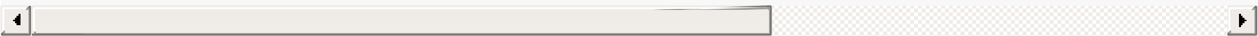
进度条组件的所有调整类可以互相组合使用。

Example



Markup

```
<div class="uk-progress uk-progress-small uk-progress-danger uk-pro
```



JAVASCRIPT组件

灯箱/Lightbox

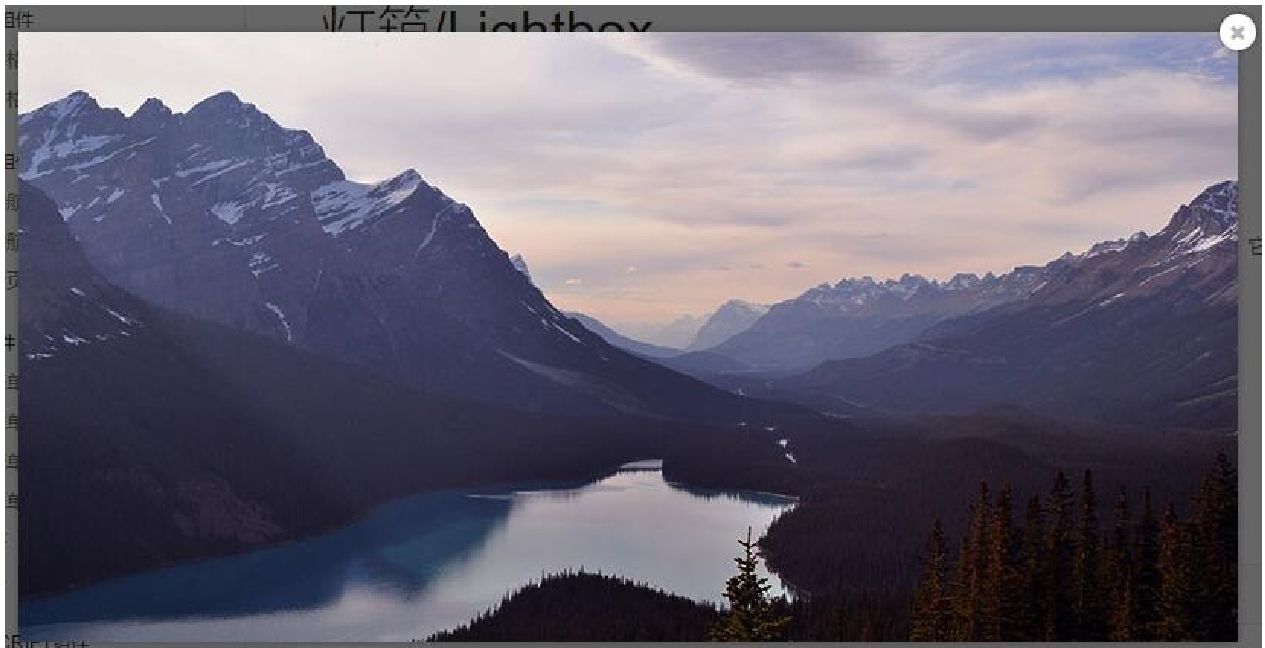
利用模态对话框为图片和视频创建一个别致的灯箱。

用法

要使用这个组件，添加 `data-uk-lightbox` 属性到一个链接到图片的锚文本中。如果title属性存在，它将会被显示为灯箱的标题。

注意 此组件需要额外添加 `lightbox.js` 文件，在 `js/components` 文件夹中。

Example



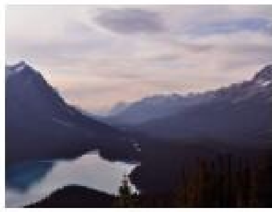
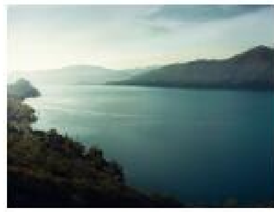
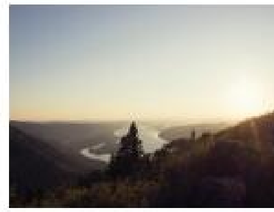
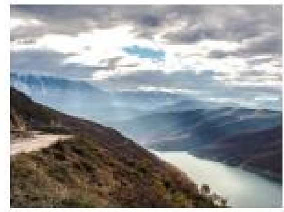
Markup

```
<a href="my-image.jpg" data-uk-lightbox title="">...</a>
```

组/Groups

你可以链接多个图片到同一个灯箱中，并在灯箱内切换显示，就是一个画廊了。只需添加 `{group:'my-group'}` 选项到每个单元的data属性，并且每个单元使用同一个grou名称，以显示在同一个灯箱中。你需要确保你已经引入了 滑动导航组件 的CSS，这样你才能在灯箱里切换显示各个单元。

Example

Markup

```
<a href="" data-uk-lightbox="{group:'my-group'}">...</a>
```

不同的内容源

灯箱并不限于图片展示。其他的媒体，比如视频，也可以显示在灯箱中，它会通过判断路径自动生成能正常的输出效果。

Example



内容类型

如果路径中没有文件扩展名，只需要添加 `data-lightbox-type="image"` 属性，视频就能正常播放了。

JavaScript 选项

选项	可用值	默认值	描述
`group`	string	false	组名用来将元素分组，作为画廊进行展示。
`duration`	integer	400	画廊切换内容时的动画持续时间。
`keyboard`	boolean	true	允许使用键盘翻页

手动初始化

```
var lightbox = UIkit.lightbox(element, { /* options */ });
```

创建动态灯箱

```
var lightbox = UIkit.lightbox.create([
                                  {'source': 'http://url/to/videc
                                  {'source': 'http://url/to/image
                              ]);

                              lightbox.show();
```

事件

名称	参数	描述
`showitem.uk.lightbox`	event, data	灯箱显示时

自动完成/Autocomplete

创建允许在输入时从预生成列表中进行选值的输入框。

用法

要使用这个组件，需要添加 `.uk-autocomplete` 类到一个包含 `input` 元素的 `<div>` 中。为了使自动完成输入框所需要的JavaScript生效，你还需要添加 `data-uk-autocomplete` 属性。添加 `{source:'PATH/TO/RESULTS'}` 到data属性中，并设置需要用JSON进行格式化的自动完成列表的路径 (示例)。 下拉菜单会被注入需要显示出来的自动完成建议。你甚至可以用键盘上的上下键来浏览下拉菜单。

注意 使用此组件需要额外添加 `autocomplete.css` 文件，在 `css/components` 文件夹中。此组件需要额外添加 `autocomplete.js` 文件，在 `js/components` 文件夹中。

Example



提示 尝试输入这些词语： Hamburg, New York, Moscow or Amsterdam.

Markup

```
<div class="uk-autocomplete uk-form" data-uk-autocomplete="{source
    <input type="text">
</div>
```

自定义模板

你还可以通过创建自定义模板使结果以不同的形式显示出来。

Example



Markup

```
<div class="uk-autocomplete uk-form" data-uk-autocomplete="{source
    <input type="text">
    <script type="text/autocomplete">
        <ul class="uk-nav uk-nav-autocomplete uk-autocomplete-resul
            {{~items}}
            <li data-value="{{ $item.value }}">
                <a>
                    {{ $item.title }}
                    <div>{{{ $item.text }}}</div>
                </a>
            </li>
            {{/items}}
        </ul>
    </script>
</div>
```

JavaScript 选项

选项	可用值	默认值	描述
`source`	url, array, callback function	[]	数据源
`minLength`	integer	3	触发自动完成的最小输入长度
`param`	string	search	发送ajax请求时的查询名称（Query name）
`delay`	integer	300	停止输入后的延时
`flipDropdown`	boolean	false	翻转显示结果的下拉菜单

手动初始化

```
var autocomplete = UIkit.autocomplete(element, { /* options */ });
```

事件

名称	参数	描述
`selectitem.uk.autocomplete`	event, data, acobject	某个值被选择时触发
`show.uk.autocomplete`	event	自动完成下拉菜单显示时触发

日期选择器/Datepicker

创建带有日期选择器的可拨动触发的下拉菜单。

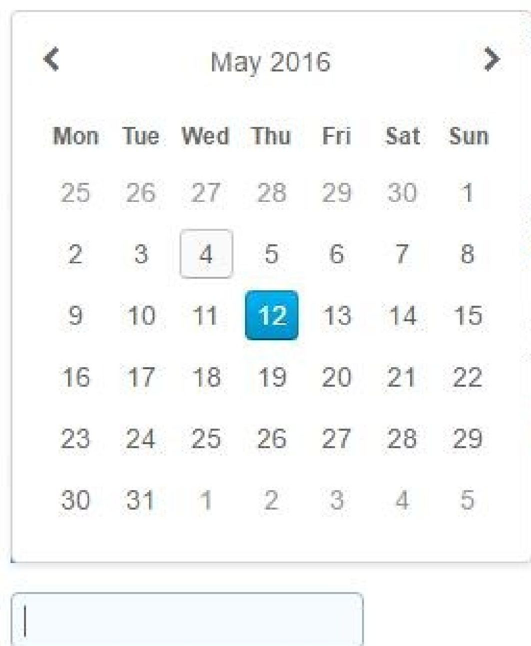
用法

创建日期选择器，只需要为 `<input>` 元素添加 `data-uk-datepicker` 属性。还可以自定义日期的格式，为 `data-uk-datepicker` 属性添加 *format* 选项就能实现。

注意 使用此组件需要额外添加 `datepicker.css` 文件，在 `css/components` 文件夹中。此组件需要额外添加 `datepicker.js` 文件，在 `js/components` 文件夹中。

日期选择器会检测 选择表单组件 的JavaScript是否被加载。这允许你在日期选择器中通过一个选择表单快捷地切换年月。

Example



Markup

```
<form class="uk-form">
    <input type="" data-uk-datepicker="{format:'DD.MM.YYYY'}">
</form>
```

JavaScript 选项

这是关于如何通过属性设置选项的例子：

```
data-uk-datepicker="{weekstart:0, format:'DD.MM.YYYY'}"
```

选项	可用值	默认值	描述
`weekstart`	integer (0..6)	1	每周开始的第一天
`i18n`	JSON object	{ months:['January',...], weekdays:['Sun',..,'Sat'] }	日期的称呼
`format`	any combination of DD, MM and YYYY	'DD.MM.YYYY'	日期格式
`offsettop`	integer	5	输入值的偏移量
`minDate`	bool (false to ignore the option)		

string (date as in `format`) integer (offset in days from current date) | false | 最小日期 | | `maxDate` | bool (false to ignore the option) string (date as in `format`) integer (offset in days from current date) | false | 最大日期 | | `pos` | 'auto', 'top', 'bottom' | 'auto' | 日期选择器出现的位置 |

手动初始化

```
var datepicker = UIkit.datepicker(element, { /* options */ });
```

事件

名称	参数	描述
`show.uk.datepicker`	event	日期选择器菜单显示时触发
`hide.uk.datepicker`	event	日期选择器菜单隐藏时触发
`update.uk.datepicker`	event	日历渲染时触发

HTML 编辑器

创建可以实时预览并且带有语法高亮的富文本编辑器或Markdown编辑器。

这个HTML编辑器可以你在输入HTML或Markdown时，生成实时预览。它包含一个工具栏，帮助你无需任何代码就能进行文本的编辑、添加链接、图片、引用和列表等。本编辑器还为HTML代码和Markdown代码提供了语法高亮，并且你还能切换全屏模式，让你可以不受任何干扰专注于内容的撰写。

用法

使用这个组件，你首先必须引入必要的 CodeMirror 和 marked 依赖。要实现它，只需要在项目的header加入适当的脚本代码就行。

注意 使用此组件需要额外添加 `htmleditor.css` 文件，在 `css/components` 文件夹中。此组件需要额外添加 `htmleditor.js` 文件，在 `js/components` 文件夹中。

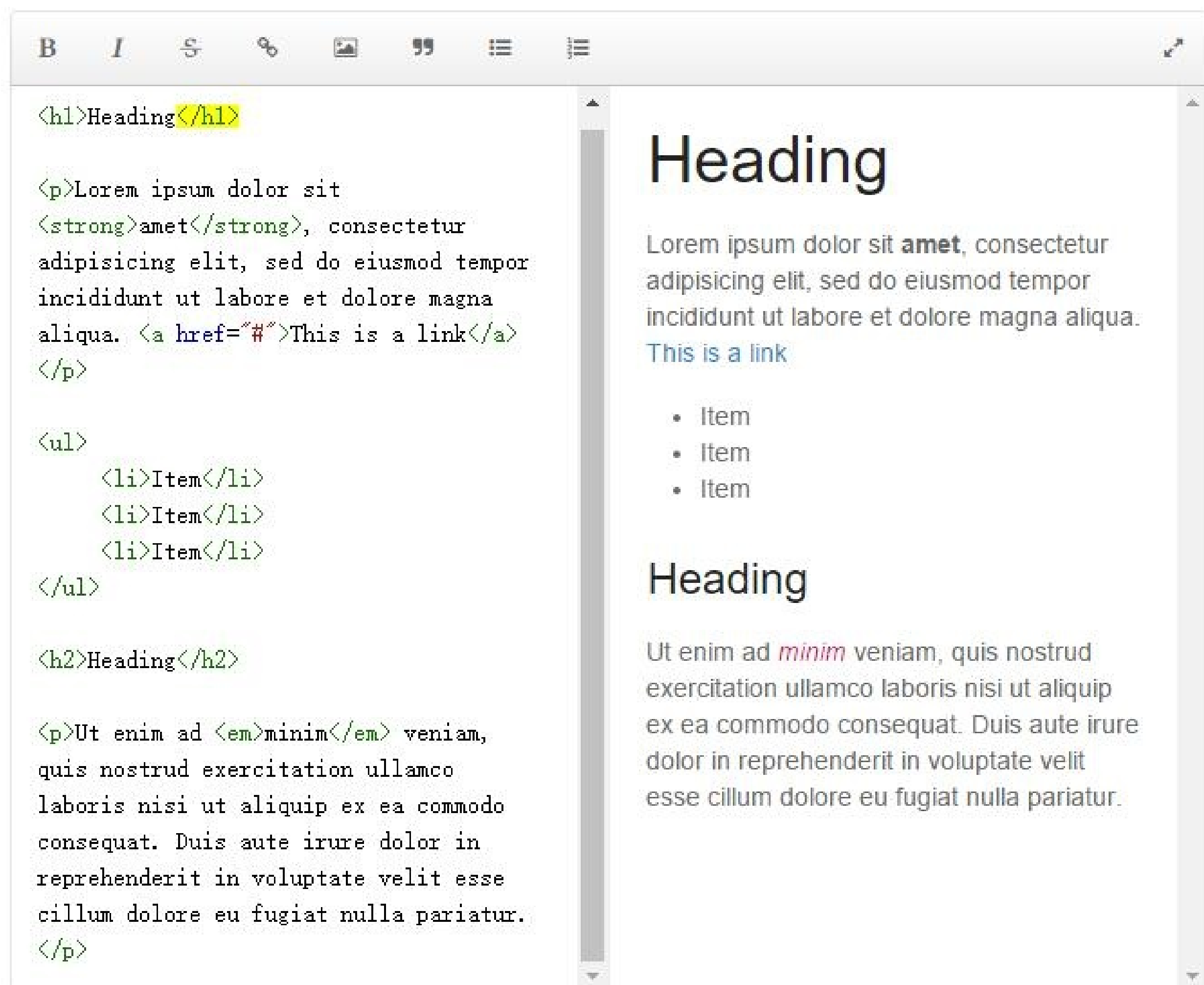
```
<!DOCTYPE html>
<html>
    <head>
        <title></title>
        <link rel="stylesheet" href="uikit.min.css" />
        <script src="jquery.js"></script>
        <script src="uikit.min.js"></script>

        <!-- Codemirror 和 marked 依赖 -->
        <link rel="stylesheet" href="codemirror/lib/codemirror.css"
        <script src="codemirror/lib/codemirror.js"></script>
        <script src="codemirror/mode/markdown/markdown.js"></script
        <script src="codemirror/addon/mode/overlay.js"></script>
        <script src="codemirror/mode/xml/xml.js"></script>
        <script src="codemirror/mode/gfm/gfm.js"></script>
        <script src="marked.js"></script>

        <!-- HTML 编辑器的 CSS 与 JavaScript -->
        <link rel="stylesheet" href="htmleditor.css">
        <script src="htmleditor.js"></script>
    </head>
    <body>
    </body>
</html>
```

最后，再为 `<textarea>` 元素添加 `data-uk-htmleditor` 属性就行了！。

Example



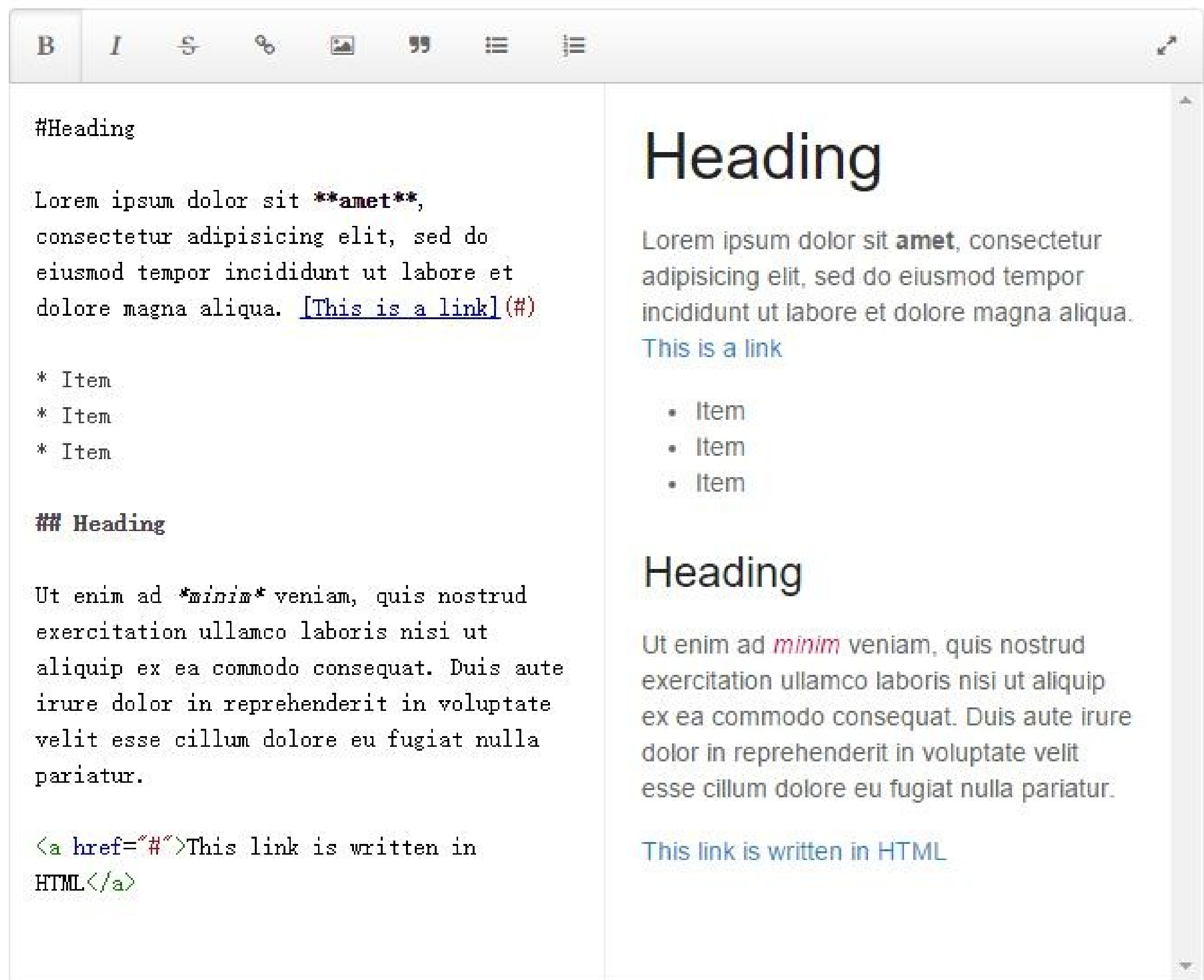
Markup

```
<textarea data-uk-htmleditor>...</textarea>
```

Markdown

还可以在HTML编辑器中编写Markdown。只需为data属性添加 `markdown:true` 选项就行。

Example



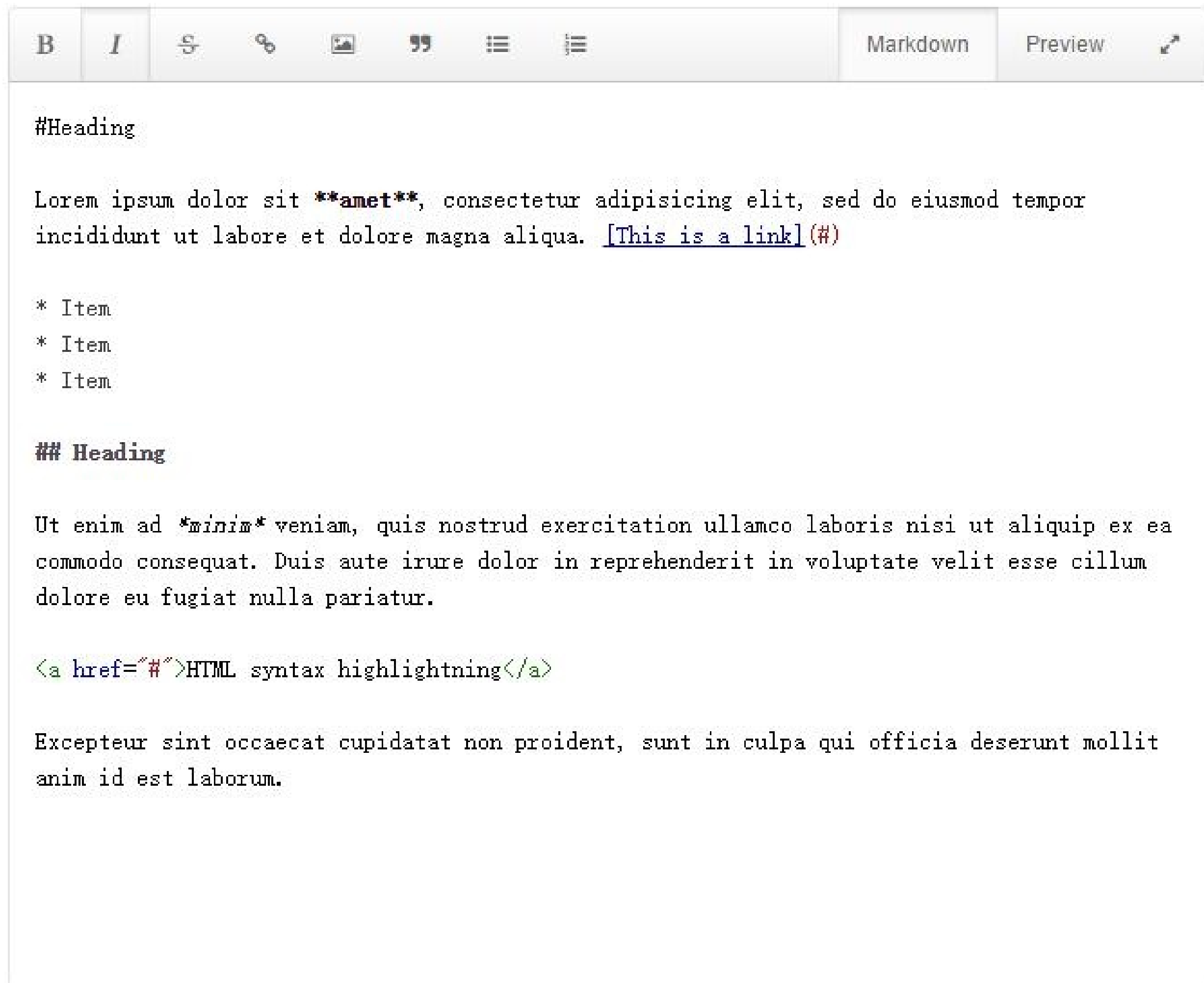
Markup

```
<textarea data-uk-htmleditor="{markdown:true}">...</textarea>
```

选项卡模式

也可以在Markdown与预览模式之间进行切换。只需添加 `data-uk-htmleditor="{mode:'tab'}` 属性。

Example



Markup

```
<textarea data-uk-htmleditor="{mode:'tab'}">...</textarea>
```

JavaScript 选项

这是如何通过属性设置选项的例子：

```
data-uk-htmleditor="{mode:'split', maxsplitsize:600}"
```

选项	可用值	默认值	描述
`mode`	'split', 'tab'	'split'	视图模式
`toolbar`	Array	["bold", "italic", "strike", "link", "picture", ...]	工具栏上显示的按钮列表
`maxsplitsize`	integer	1000	触发由分割模式切换到选项卡模式的响应式行为的最小浏览器视口宽度。
`lblPreview`	任意 string	'Preview'	预览模式的标签字符串（Label string）
`lblCodeview`	任意 string	'Markdown'	代码模式的标签字符串（Label string）

手动初始化

```
var htmleditor = UIkit.htmleditor(textarea, { /* options */ });
```

滚动条/Slider

创建一个可以作为响应式旋转木马滚动条的条目列表。

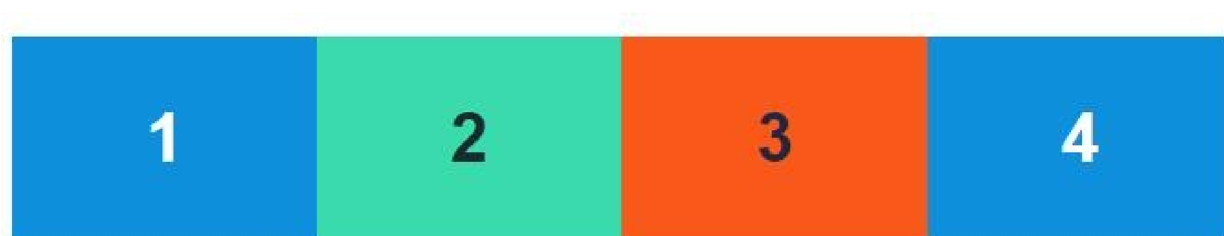
滚动条能响应式地显示元素，可以通过鼠标和触摸手势滚动。

用法

要使用滚动条组件，需要为包含 `.uk-slider-container` 元素的容器元素添加 `data-uk-slider` 属性。添加条目的列表，并为此列表添加 `.uk-slider` 类。使用 `.uk-width-*` 和 `.uk-grid-width-*` 类以确定每次显示多少个元素。

注意 使用此组件需要额外添加 `slider.css` 文件，在 `css/components` 文件夹中。此组件需要额外添加 `slider.js` 文件，在 `js/components` 文件夹中。

Example



Markup

```
<div data-uk-slider>

    <div class="uk-slider-container">
        <ul class="uk-slider uk-grid-width-medium-1-4">
            <li>...</li>
            ...
        </ul>
    </div>

</div>
```

导航/Navigation

滚动条本身可以使用鼠标点击和拖拽或者触屏设备中滑动进行滚动。添加一个可以点击的滑动导航也是个好主意。滑动导航添加的箭头按钮会在鼠标悬停在滑动条上时显示出来。

Example



```
<div class="uk-slidenav-position" data-uk-slider>

    <div class="uk-slider-container">
        <ul class="uk-slider uk-grid-width-medium-1-4">
            <li>...</li>
            ...
        </ul>
    </div>

    <a href="" class="uk-slidenav uk-slidenav-contrast uk-slidenav-
    <a href="" class="uk-slidenav uk-slidenav-contrast uk-slidenav-

</div>
```

居中模式/Center Mode

默认情况下，滚动条的元素总是对齐滚动条容易的左边缘。如果你想让元素居中，设置 `center` 属性的值为 `true` 即可。

注意 被居中的列表元素通常会被添加 `.uk-active` 类。如果你想高亮显示居中的元素，添加该类名即可。

Example



Markup

```
<div data-uk-slider="{center:true}">

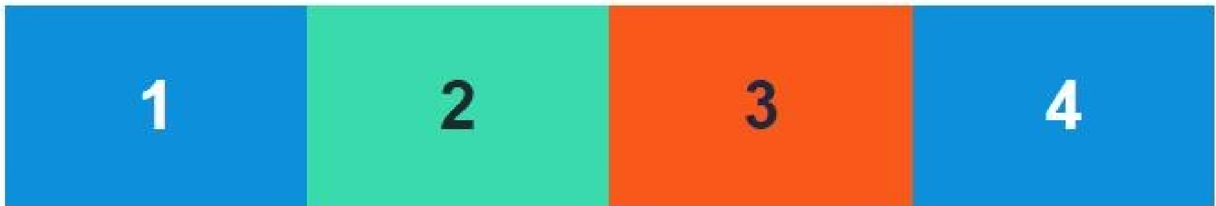
    <div class="uk-slider-container">
        <ul class="uk-slider uk-grid-width-medium-1-4">
            <li>...</li>
            ...
        </ul>
    </div>

</div>
```

禁用无限滚动

默认情况下，滚动条循环显示所有条目。要禁用这种行为，设置 `infinite` 参数为 `false` 即可。在启用或禁用居中模式中都同样有效。

Example



Markup

```
<div data-uk-slider="{infinite: false}">

    <div class="uk-slider-container">
        <ul class="uk-slider uk-grid-width-medium-1-4">
            <li>...</li>
            ...
        </ul>
    </div>

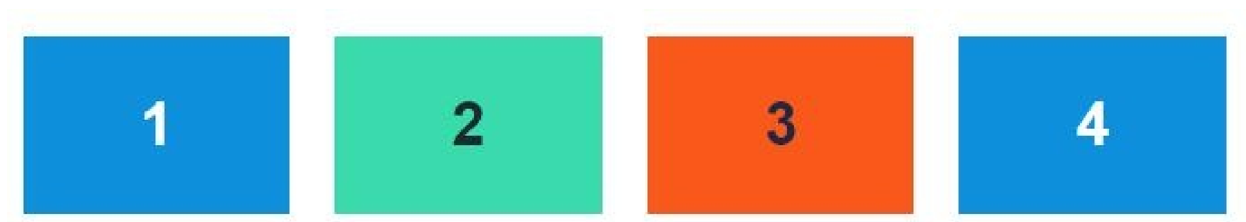
</div>
```

条目排水沟/Item gutter

如果你想为滚动条中的元素添加间距，添加 `.uk-grid` 类到滚动条容器即可。元素将会根据网格排水沟尺寸被分隔开。

注意 你可以使用网格的修饰类 `uk-grid-medium` 和 `uk-grid-small` 来调整排水沟的尺寸。

Example



条目宽度/Item width

如果要设置滚动条中各元素的宽度，使用 UIkit 网格中的宽度类即可。既可以在滚动条容器上使用 `uk-grid-width-*` 类名，也可以为列表中每个条目单独使用 `uk-width-*` 类名。

Example: 为条目单独设置宽度



Markup

```
<div data-uk-slider>

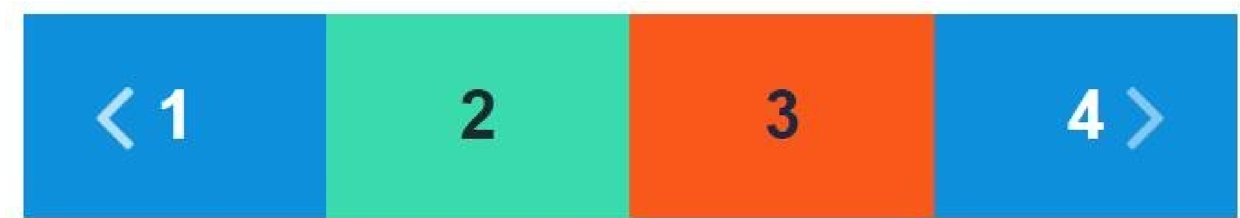
    <div class="uk-slider-container">
        <ul class="uk-slider">
            <li class="uk-width-1-3">...</li>
            <li class="uk-width-1-5">...</li>
            <li class="uk-width-2-5">...</li>
            ...
        </ul>
    </div>

</div>
```

响应式行为

为了适应不同的视口，你可以使用网格的响应式类名。在下面的例子中，滚动条会在大视口中显示4个条目，在中视口中显示3个，在小视口中显示一个。

Example: 响应式宽度

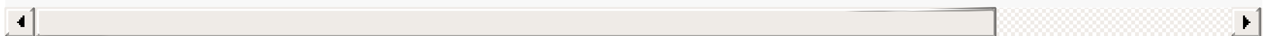


Markup

```
<div data-uk-slider>

    <div class="uk-slider-container">
        <ul class="uk-slider uk-grid-width-medium-1-3 uk-grid-widt
            <li>...</li>
            ...
        </ul>
    </div>

</div>
```



全屏模式

滚动条包含全屏模式，在全屏模式下，滚动条将延伸至100%的视口高度。

Markup

```
<div data-uk-slider>

    <div class="uk-slider-container">
        <ul class="uk-slider uk-slider-fullscreen">
            <li>...</li>
            ...
        </ul>
    </div>

</div>
```

JavaScript 选项

选项	可用值	默认值	描述
`center`	boolean	false	条目居中模式
`threshold`	boolean	true	移动鼠标触发元素拖动的阈值，以像素为单位。
`infinite`	boolean	true	无限滚动
`activecls`	string	uk-active	在居中模式中，添加到被选中条目上的类名。
`autoplay`	false	boolean	是否让滚动条的内容条目自动切换
`pauseOnHover`	boolean	true	鼠标悬停在滚动条上时暂停播放
`autoplayInterval`	integer	7000	切换滚动条内容条目的时间间隔

手动初始化元素

```
var slider = UIkit.slider(element, { /* options */ });
```

事件/Events

事件名	**Parameter**	描述
`focus.uikit.slider`	event, index, item	条目获得焦点时触发

滑块集/Slideset

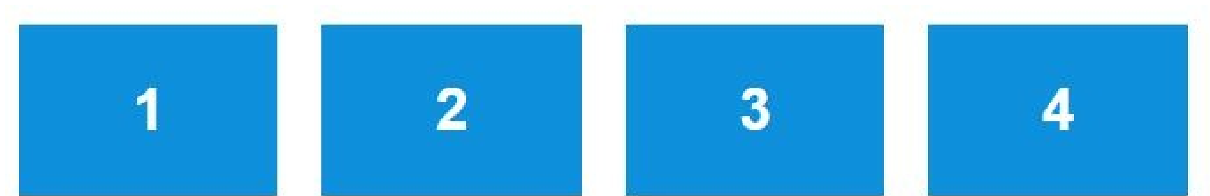
创建条目的组和集合，并允许循环显示。

用法

使用这个组件需要为容器元素添加 `data-uk-slideset` 属性。添加 `default` 选项到 `data-uk-slideset` 属性，调整一个滑块集中条目的个数。

注意 此组件需要额外添加 `slideset.js` 文件，在 `js/components` 文件夹中。

Example



Markup

```
<div data-uk-slideset="{default: 4}">
    <ul class="uk-slideset">
        <li><img src="" alt=""></li>
        <li><img src="" alt=""></li>
        ...
    </ul>
</div>
```

导航/Navigations

为滑块集添加导航，只需使用 `uk-slideset-nav` 类。它将动态地创建一个基于现有滑块数量的导航。

Markup

```
<div data-uk-slideset="{default: 4}">
    <ul class="uk-slideset">
        <li>...</li>
        <li>...</li>
    </ul>
    <ul class="uk-slideset-nav">...</ul>
</div>
```

要切换相邻的滑块，使用 `data-uk-slideset-item` 属性，并设置该属性的值为 `next` 和 `previous` 。这些带有 `data-uk-slideset-item` 属性的元素必须放在带有 `data-uk-slideset` 属性的容器中。

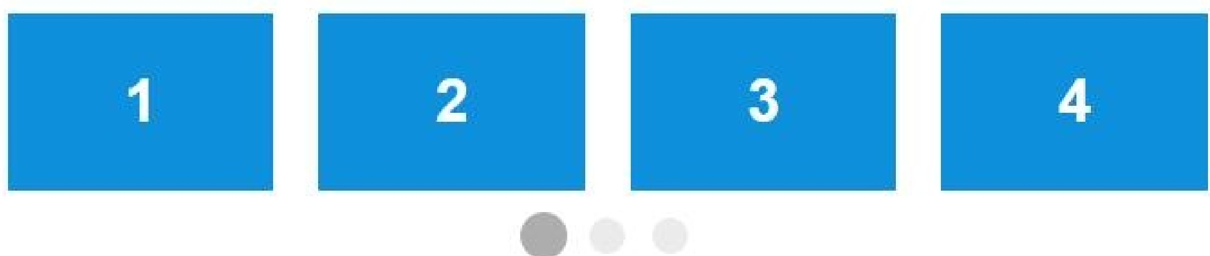
Markup

```
<div data-uk-slideset="{default: 4}">
    <ul class="uk-slideset">
        <li>...</li>
        <li>...</li>
    </ul>
    <a href=""data-uk-slideset-item="previous"></a>
    <a href=""data-uk-slideset-item="next"></a>
</div>
```

滑动导航和圆点导航/Slidenav and Dotnav

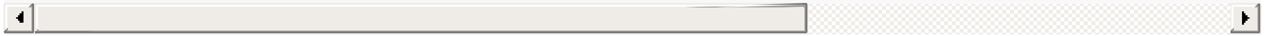
你可以使用 UIkit 的其他任意组件来做滑块集的导航。比如 Slidenav 和 Dotnav 可以做出像下面例子这样的导航。

Example



Markup

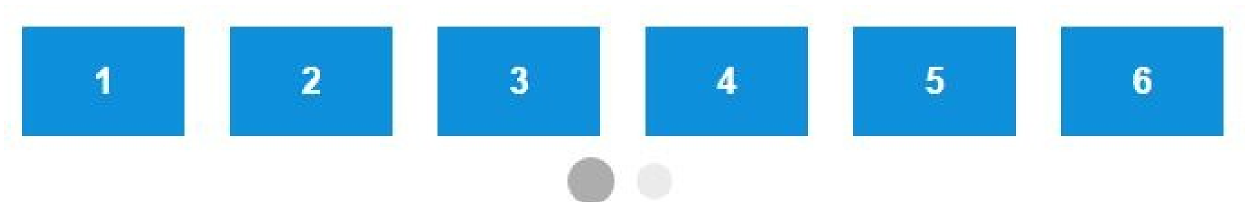
```
<div data-uk-slideset="{default: 4}">
    <div class="uk-slidenav-position">
        <ul class="uk-slideset">
            <li>...</li>
            <li>...</li>
        </ul>
        <a href="" class="uk-slidenav uk-slidenav-previous" data-uk
        <a href="" class="uk-slidenav uk-slidenav-next" data-uk-sli
    </div>
    <ul class="uk-slideset-nav uk-dotnav uk-flex-center">...</ul>
</div>
```



响应式条目/Responsive items

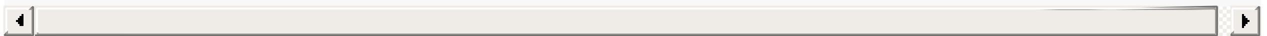
滑块集支持基于条目可见性的媒体查询。只需添加一个断点选项到 `data-uk-slideset` 属性中，比如 `small`，`medium`，`large` 。并设置在该断点宽度以上的视口中你想要显示的条目数，如下面例子所示：

Example



Markup

```
<div data-uk-slideset="{small: 2, medium: 4, large: 6}">...</div>
```

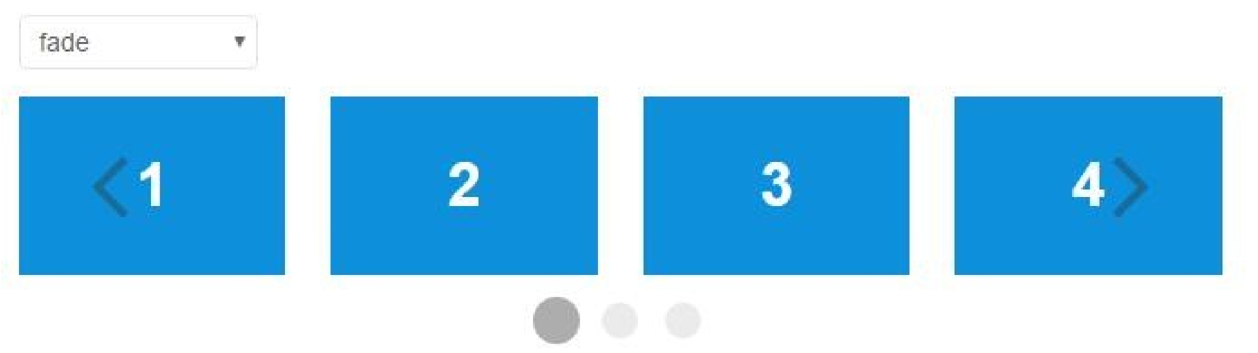


动画/Animations

这里有那么多的动画可以用在滑块集里，用来显示下一批滑块。只要为 `data-uk-slideset` 添加 `animation` 选项并设置你想要的动画。动画的持续时间可以通过设置 `duration` 选项来修改。

动画	描述
`fade`	淡入淡出
`scale`	缩放
`slide-horizontal`	水平滑入滑出
`slide-vertical`	垂直滑入滑出
`slide-top`	从顶部滑出，从顶部滑入
`slide-bottom`	从底部滑出，从底部滑入

Example - 在下拉框里选你想要预览的



Markup

```
<div data-uk-slideset="{animation: 'scale', duration: 200}">...</di
```

过滤/Filter

你还可以使用过滤来实现只显示滑块集中特定的条目。为此，需要为每个过滤控件添加 `data-uk-filter` 属性来规定它们过滤的分类。然后还要为每个滑块条目添加 `data-uk-filter` 属性来规定它们各自属于哪个分类。

注意 过滤控件可以放在滑块集中，或者，为过滤控件添加 `controls` 选项到 `data-uk-slideset` 属性中，并将滑块集的 id 设置在该选项中。如下面的例子：

Example



Markup

```
<!-- 过滤控件包含在滑块集中 -->
<div data-uk-slideset>
    <ul>
        <li data-uk-filter=""><a></a></li>
        <li data-uk-filter="filter-a"><a></a></li>
        <li data-uk-filter="filter-b"><a></a></li>
    </ul>

    <ul class="uk-slideset">
        <li data-uk-filter="filter-a"><img src=""></li>
        <li data-uk-filter="filter-b"><img src=""></li>
    </ul>
</div>

<!-- 过滤控件在滑块集外面 -->
<ul id="my-id">
    <li data-uk-filter=""><a></a></li>
    <li data-uk-filter="filter-a"><a></a></li>
    <li data-uk-filter="filter-b"><a></a></li>
</ul>

<div data-uk-slideset"{controls: '#my-id'}">
    <ul class="uk-slideset">
        <li data-uk-filter="filter-a"><img src=""></li>
        <li data-uk-filter="filter-b"><img src=""></li>
    </ul>
</div>
```

JavaScript 选项

选项	可用值	默认值	描述
`default`	integer	1	一个滑块集中默认可见条目数
`small`	integer	null	小视口（small）中显示的条目数
`medium`	integer	null	中视口（medium）中显示的条目数
`large`	integer	null	大视口（large）中显示的条目数
`xlarge`	integer	null	特大视口（Xlarge）中显示的条目数
`animation`	string	'fade'	动画的名字
`duration`	integer	200	以毫秒为单位的动画持续时间
`delay`	integer	100	一个滑块集中多个条目之间的动画延时。
`filter`	string	''	滑块条目过滤
`autoplay`	Boolean	false	定义滑块集条目是否自动播放
`pauseOnHover`	Boolean	true	鼠标悬停在滑块集上时，赞同自动播放
`autoplayInterval`	integer	7000	切换滑块集条目的时间间隔

手动初始化

```
var slideset = UIkit.slideset(element, { /* options */ });
```

事件

名称	参数	描述
`show.uk.slideset`	event, set	滑块集显示时触发

幻灯片/Slideshow

创建可以全屏模式和遮罩效果的，具有炫酷过渡效果的响应式图片或视频幻灯片。

用法

要创建幻灯片，只需要添加 `.uk-slideshow` 类到一个 `<ul>` 元素中，并将你的图片放入列表条目内。为了加载必要的JavaScript，还需要添加 `data-uk-slideshow` 属性。

注意 使用此组件需要额外添加 `slideshow.css` 文件，在 `css/components` 文件夹中。此组件需要额外添加 `slideshow.js` 文件，在 `js/components` 文件夹中。

Example

注意 如果需要自动播放，为data属性添加 `{autoplay:true}` 选项就行了。

Markup

```
<ul class="uk-slideshow" data-uk-slideshow="{autoplay:true}">
    <li><img src="" width="" height="" alt=""></li>
</ul>
```

导航/Navigations

在幻灯片中进行导航，使用 `data-uk-slideshow-item` 属性就行。为了能指向幻灯片，你必须为每个导航单元设置data属性指向每个幻灯片单元的序号。带有 `data-uk-slideshow-item` 属性的元素需要被放置在带有 `data-uk-slideshow` 的容器中。

Markup

```
<div data-uk-slideshow>
    <ul class="uk-slideshow">
        <li></li>
        <li></li>
    </ul>
    <li data-uk-slideshow-item="0"><a href="">...</a></li>
    <li data-uk-slideshow-item="1"><a href="">...</a></li>
</div>
```

将data属性设置为 `next` and `previous` 就能进行相邻幻灯片之间的切换。像这样：

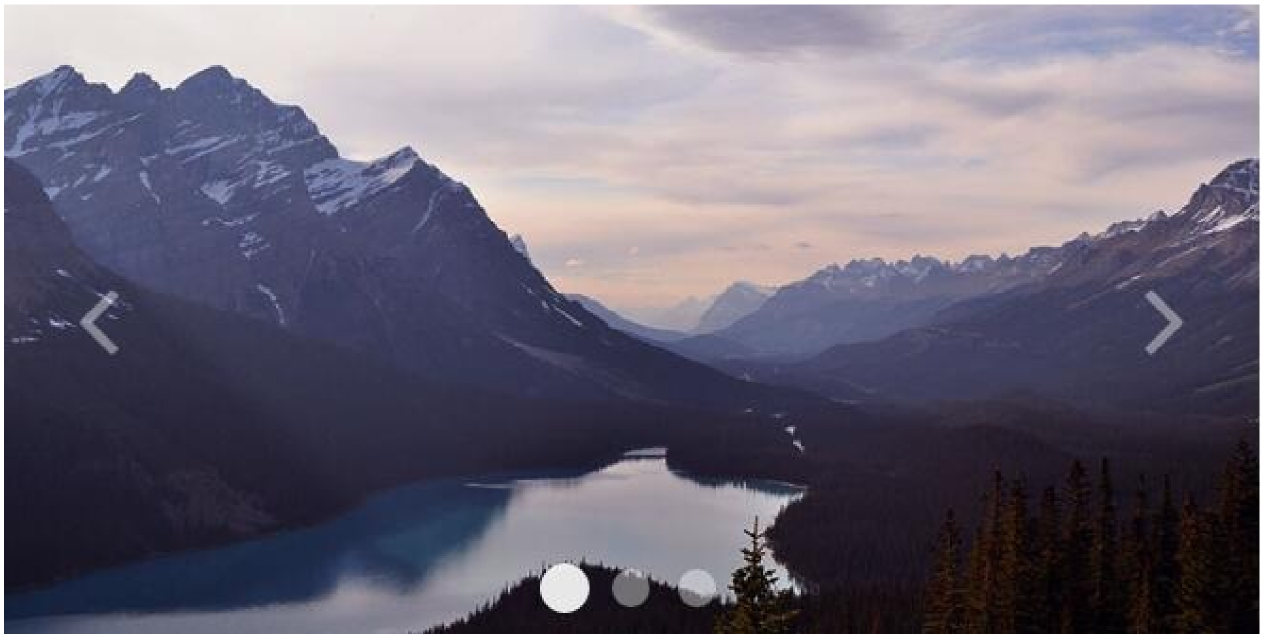
Markup

```
<div data-uk-slideshow>
    <ul class="uk-slideshow">
        <li></li>
        <li></li>
    </ul>
    <a href="" data-uk-slideshow-item="previous"></a>
    <a href="" data-uk-slideshow-item="next"></a>
</div>
```

滑动导航和圆点导航

幻灯片组件的灵活性使之可以用任何其他UIkit组件进行幻灯片的导航。比如滑动导航和圆点导航可以用来像下面这样作为幻灯片的导航。

Example



Markup

```
<div class="uk-slidenav-position" data-uk-slideshow>
    <ul class="uk-slideshow">
        <li></li>
        <li></li>
    </ul>
    <a href="" class="uk-slidenav uk-slidenav-contrast uk-slidenav-
    <a href="" class="uk-slidenav uk-slidenav-contrast uk-slidenav-
    <ul class="uk-dotnav uk-dotnav-contrast uk-position-bottom uk-t
        <li data-uk-slideshow-item="0"><a href=""></a></li>
        <li data-uk-slideshow-item="1"><a href=""></a></li>
    </ul>
</div>
```

过渡形式

幻灯片组件允许添加不同的幻灯片切换效果。你需要做的就是将 `animation` 参数添加到data属性中，并声明你希望使用的动画。查看下面的表格了解咱都提供了哪些动画效果。

Class	描述
`fade`	淡入
`scroll`	滚动进入
`scale`	放大
`swipe`	滑动进入、滑动离开

要使用下面这些增强的过渡效果，必须在文档head中引入 `slideshow-fx.js` 。下面表格不翻译了，看后面的演示吧。

Class	描述
`slice-down`	The items slide down in slices.
`slice-up`	The items slide up in slices.
`slice-up-down`	The sliced items slide in alterning directions.
`fold`	The items are folded in.
`puzzle`	The items are divided in squares that randomly fade in.
`boxes`	The items are divided in squares that scale in diagonally from the top left corner.
`boxes-reverse`	The items are divided in squares that scale in diagonally from the bottom right corner.
`random-fx`	Different animations are applied randomly.

Markup

```
<ul class="uk-slideshow" data-uk-slideshow="{animation: 'random-fx
```

Ken Burns 效果

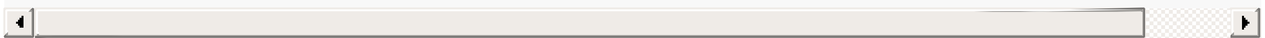
炫酷的 Ken Burns 效果也能用在幻灯片里。只需要添加 `{kenburns:true}` 选项到data属性中就行。这个效果还能用在视频上。

Example



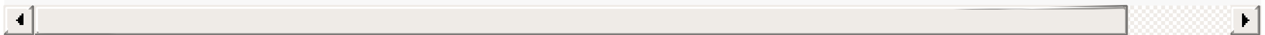
Markup

```
<ul class="uk-slideshow" data-uk-slideshow="{kenburns:true}">...</u
```



自定义动画持续时间:

```
<ul class="uk-slideshow" data-uk-slideshow="{kenburns:'30s'}">...</
```

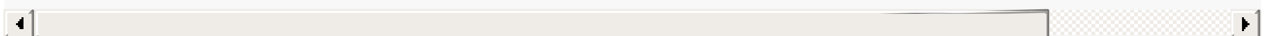


全屏幻灯片

创建填满整个视口的全屏幻灯片，只需要添加一个 `.uk-slideshow-fullscreen` 类。

Markup

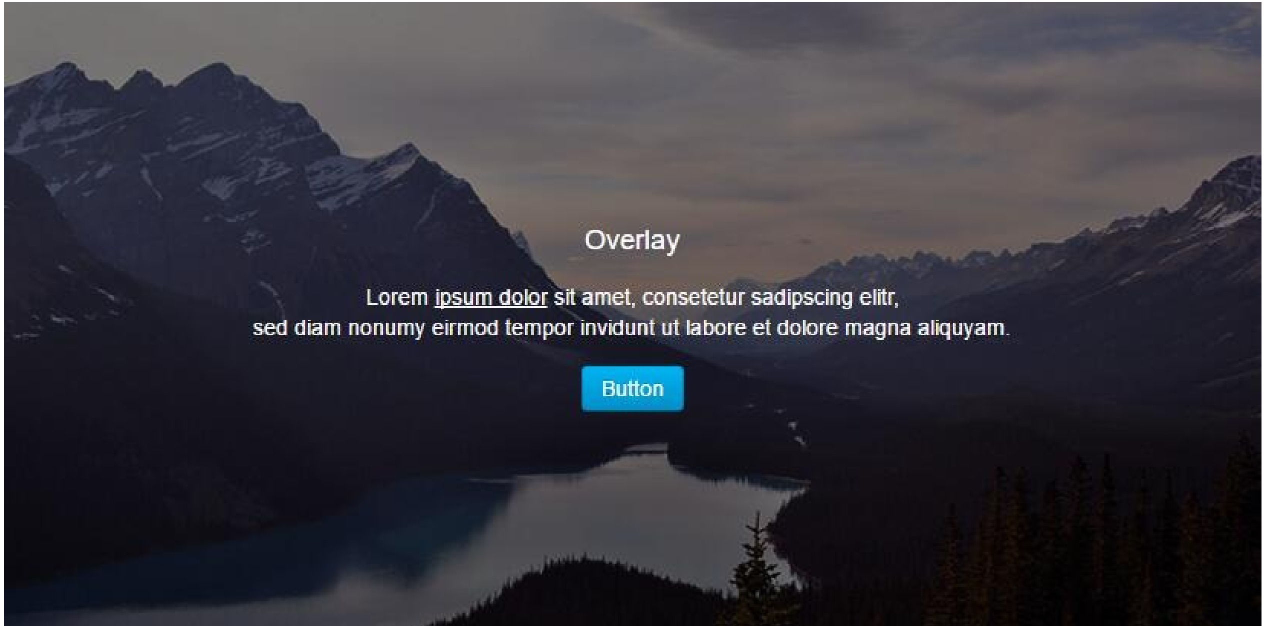
```
<ul class="uk-slideshow uk-slideshow-fullscreen" data-uk-slideshow>
```



遮罩

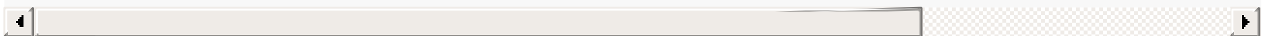
还可以用遮罩来提升幻灯片的效果，为幻灯片列表条目内的 `<div>` 元素添加 `.uk-uk-overlay-panel` 类就行。添加 `.uk-overlay-background` 和 `.uk-overlay-fade` 带来背景与过渡效果。要实现幻灯片显示时即触发遮罩，添加 `.uk-uk-overlay-active` 到列表容器即可。更多选项查看 遮罩组件.

Example



Markup

```
<ul class="uk-slideshow uk-overlay-active" data-uk-slideshow>
    <li>
        <img src="" width="" height="" alt="">
        <div class="uk-overlay-panel uk-overlay-background uk-overl
    </li>
</ul>
```



视频

UIkit 允许在幻灯片中放入视频和iframe。

Example

Markup

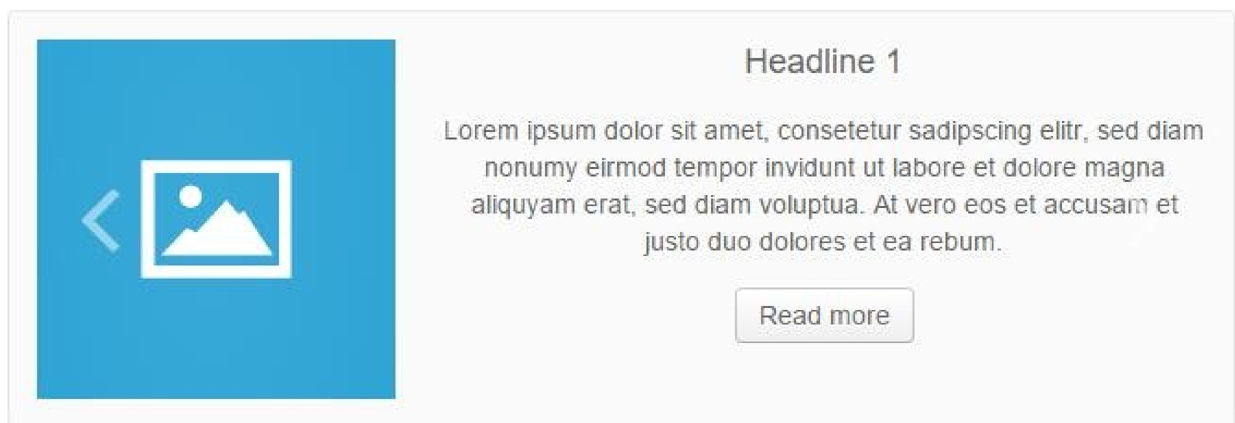
```
<!-- 带有视频的幻灯片 -->
<ul class="uk-slideshow">
    <li>
        <video width="" height="" autoplay loop muted controls>
            <source src="" type="">
        </video>
    </li>
</ul>

<!-- 带有Iframe的幻灯片 -->
<ul class="uk-slideshow">
    <li>
        <iframe src="" width="" height="" frameborder="0" allowfull
    </li>
</ul>
```

内容

基本上可以在幻灯片内插入任何内容，比如文本甚至整个网格。

Example



JavaScript 选项

选项	默认值	描述
`animation`	'fade'	定义幻灯片之间优先呈现的过渡效果
`duration`	500	过渡效果持续时间
`height`	'auto'	幻灯片高度
`start`	0	定义显示的第一张幻灯片
`autoplay`	false	是否自动切换幻灯片
`pauseOnHover`	true	鼠标悬停时是否暂停自动播放
`autoplayInterval`	7000	定义幻灯片切换的时间跨度
`videoautoplay`	true	定义是否自动开始播放视频
`videomute`	true	定义是否静音播放视频
`kenburns`	false	定义是否启用 Ken Burns 效果
`kenburnsanimations`	uk-animation-middle-left, uk-animation-top-right,	

uk-animation-bottom-left, uk-animation-top-center, uk-animation-bottom-right | 动画方向 || `slices` | 15 | 定义碎片的数量，如果启用了 "Slice" 过渡效果的话。 |

事件

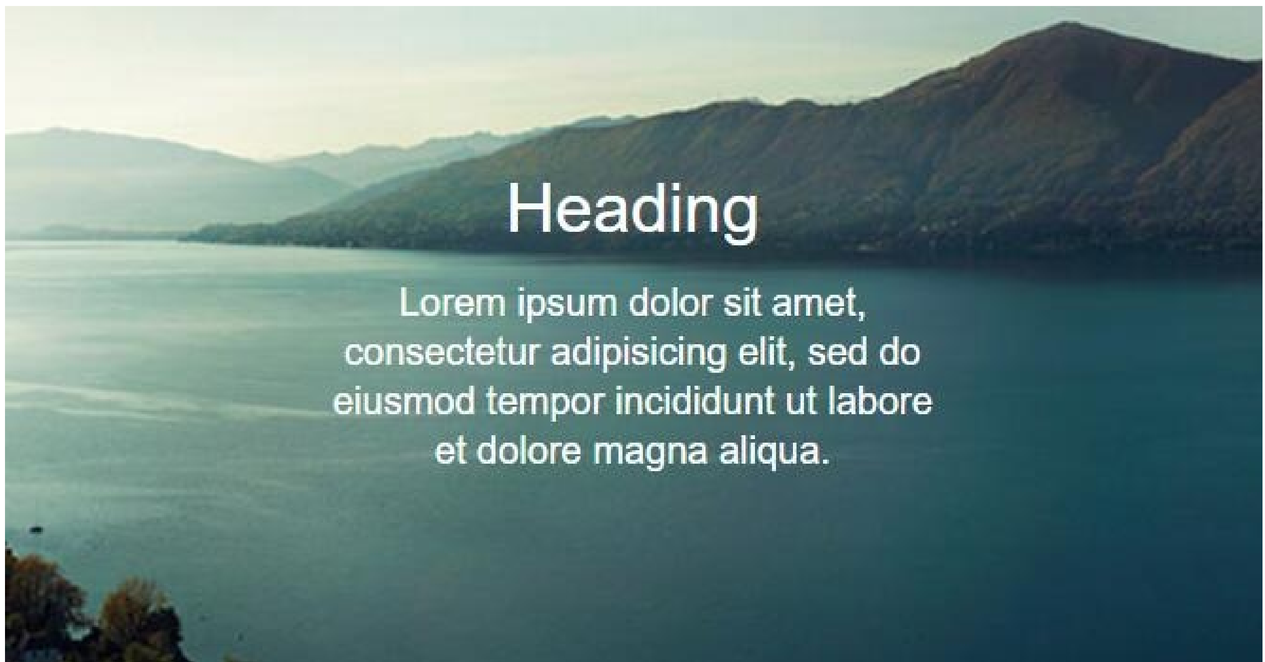
名称	参数	描述
`show.uk.slideshow`	event, next slide, current slide	新的一页显示时触发（动画完成后）
`beforeshow.uk.slideshow`	event, next slide, current slide	新的一页显示前触发（动画完成前）

视差/Parallax

依赖于页面滚动条位置的动态 CSS 特性。

用法

要使用这个组件，需要添加 `data-uk-parallax` 属性到容器元素。并为每个你想要使之动态变化的 CSS 属性添加对应的选项值。注意 此组件需要额外添加 `parallax.js` 文件，在 `js/components` 文件夹中。



Markup

```
<div data-uk-parallax="{bg: '-200'}">...</div>
```

选项

UIkit 提供了一系列的选项，你可以把它们添加到 `data-uk-parallax` 属性中：

选项	描述
`x`	以像素为单位的 X 轴方向位移。
`xp`	以百分比为单位的 X 轴方向位移。
`y`	以像素为单位的 Y 轴方向位移。
`yp`	以百分比为单位的 Y 轴方向位移。
`bg`	使背景图片动态变化。
`bgp`	以百分比为单位的背景图片动态变化。
`rotate`	以度为单位的顺时针动态变化。
`scale`	缩放的动态变化
`color`	色彩的动态变化（需要设置起始值和终止值）
`background-color`	背景色彩的变化（需要设置起始值和终止值）
`border-color`	border 色彩的变化（需要设置起始值和终止值）
`opacity`	透明度的变化

注意 你可以基础性地使任意只有一值的 CSS 属性动态变化，比如宽度和高度，直接添加到属性中即可。

Markup

```
<div data-uk-parallax="{y: '-200', opacity: '0'}">...</div>
```

起始值和终止值

属性通常由当前值变化到你设置的目标值。然而，你还可以自己定义一个起始值。这将通过传递字符串到包含由逗号隔开的两个值的选项中来实现。

注意 某些属性，比如颜色，必须要有起始值和终止值。

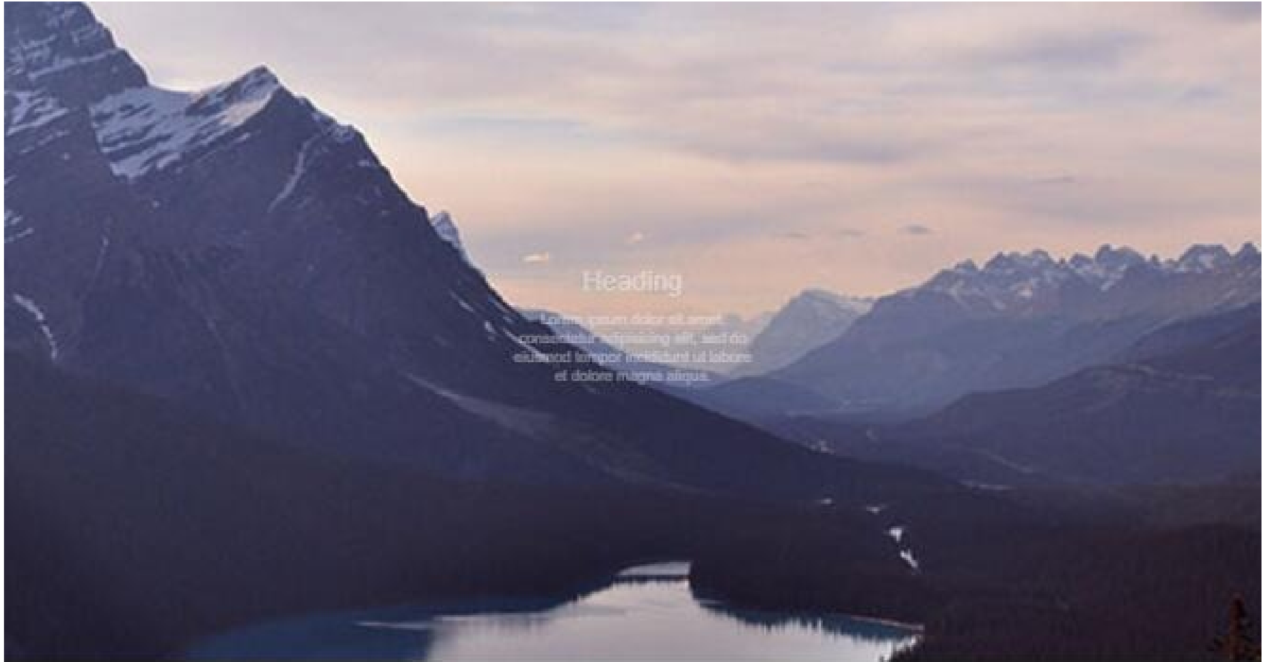
Markup

```
<div data-uk-parallax="{x: '-100,100', 'background-color': '#EBF7FI
```

嵌套的动画/Nested animation

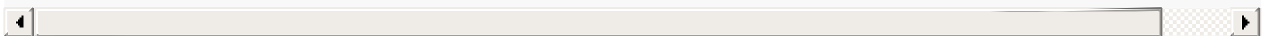
为前提的元素使用不同的动画是很简单的事情。只需在视差容器中再创建一个容器，并在新的 `data-uk-parallax` 属性中添加你的选项即可。

Example



Markup

```
<div data-uk-parallax="{bg: -200}">
    <div data-uk-parallax="{opacity: '0,1', scale: '0,1'}">...</di
</div>
```



目标选项/Target Option

通常，视差动画从元素进入视口开始，到离开视口结束。开始和终止动画基于另一个元素在视口中的可见性，使用 `target` 选项进行设置。在使用嵌套动画时会很有帮助。

Example



Markup

```
 <div id="target-id">...</div>
<div data-uk-parallax="{target: '#target-id'}">...</div>
```

速度/Velocity

添加 `velocity` 选项，调整动画的缓动效果。

Example



Markup

```
<div data-uk-parallax="{velocity: '0.5'}">
    ...
</div>
```

视口/Viewport

使用 `viewport` 选项，动画持续时间将被调整。其值为 `1` 或 `false` 时，视差动画从元素进入视口开始，到离开视口结束。将其设置为 `0.5` ，如下面的例子，动画只在元素出现的前半个视口中发生。

Example



Markup

```
<div data-uk-parallax="{viewport: '0.5'}">...</div>
```

JavaScript 选项

选项	可用值	默认值	描述
`velocity`	float	0.5	页面滚动时，动画的速度
`target`	mixed	false	关于动画持续时间的元素尺寸参考/Element dimension reference for animation duration.
`viewport`	float (0 to 1)	false	依赖于视口的动画范围
`media`	integer / string	false	启用视差效果的视口宽度条件（比如640px），或CSS媒体查询

手动初始化

```
var parallax = UIkit.parallax(element, { /* options */ });
```

手风琴/Accordion

创建一个列表菜单，通过点击它们的标题来展开或收缩内容。

用法

使用手风琴菜单，为容器元素添加 `uk-accordion` 类和 `data-uk-accordion` 属性就行。然后为容器内的每项内容添加 `uk-accordion-content` 类。最后，为标题或者其他元素添加 `uk-accordion-title` 类来创建拨动器。注意 使用此组件需要额外添加 `accordion.css` 文件，在 `css/components` 文件夹中。此组件需要额外添加 `accordion.js` 文件，在 `js/components` 文件夹中。

Example

Heading 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.

Heading 2

Heading 3

Markup

```
<div class="uk-accordion" data-uk-accordion>

    <h3 class="uk-accordion-title">...</h3>
    <div class="uk-accordion-content">...</div>

    <h3 class="uk-accordion-title">...</h3>
    <div class="uk-accordion-content">...</div>

    <h3 class="uk-accordion-title">...</h3>
    <div class="uk-accordion-content">...</div>

</div>
```

多条同时展开

同时显示多条内容，而不在其他内容显示时被隐藏，添加 `{collapse: false}` 选项到data属性就行了。

Example

Heading 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.

Heading 2

Heading 3

JavaScript 选项

选项	可用值	默认值	描述
`showfirst`	boolean	true	初始化时首先显示
`collapse`	boolean	true	允许同时展开多条
`animate`	boolean	true	拨动动画
`easing`	string	swing	动画功能
`duration`	integer	300	动画持续时间
`toggle`	string	.uk-accordion-title	拨动器的Css选择器
`containers`	string	.uk-accordion-content	内容容器的Css选择器
`clsactive`	string	uk-active	条目被激活时添加这个Class

手动初始化

```
var accordion = UIkit.accordion(element, { /* options */ });
```

事件

名称	参数	描述
`toggle.uk.accordion`	event, active, toggle, content	拨动时触发

通知/Notify

创建可以被拨动显示的通知，并且能自动淡出。

当你将鼠标悬停在通知上面时，它不会自动消失，直到你移开鼠标。还可以通过点击关闭通知。

用法

通知组件提供了一个简单的API，你可以在应用程序代码中重用它。下面的JavaScript代码片段让你快速上手。注意 使用此组件需要额外添加 `notify.css` 文件，在 `css/components` 文件夹中。此组件需要额外添加 `notify.js` 文件，在 `js/components` 文件夹中。

JavaScript

```
UIkit.notify({
    message : 'Bazinga!',
    status  : 'info',
    timeout : 5000,
    pos     : 'top-center'
});

// Shortcuts
UIkit.notify('My message');
UIkit.notify('My message', status);
UIkit.notify('My message', { /* options */ });
```

Example

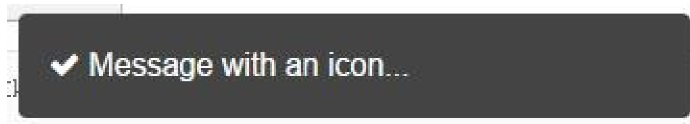



```
UIkit.notify("Message...");
```

HTML 信息

可以在通知信息内嵌HTML，比如 图标 。

Example




```
UIkit.notify("<i class='uk-icon-check'></i> Message with an icon..
```

延迟和常驻/Delay and sticky

你可以通过 `timeout` 定义在多少毫秒内通知消息是可见的。还可以把延迟设为0，让通知消息常驻不消失。

Example




```
UIkit.notify("Message...", {timeout: 0});
```

定位

添加以下参数中的一个来调整通知出现的位置。

`top-center`

```
UIkit.notify("...", {pos:'top-center'})
```

`top-left`

```
UIkit.notify("...", {pos:'top-left'})
```

`top-right`

```
UIkit.notify("...", {pos:'top-right'})
```

`bottom-center`

```
UIkit.notify("...", {pos:'bottom-center'})
```

`bottom-left`

```
UIkit.notify("...", {pos:'bottom-left'})
```

`bottom-right`

```
UIkit.notify("...", {pos:'bottom-right'})
```

状态

通知可以通过添加一些状态效果来表达一般的消息，成功的消息，警告或者危险等等。

`info`

```
UIkit.notify("...", {status:'info'})
```

`success`

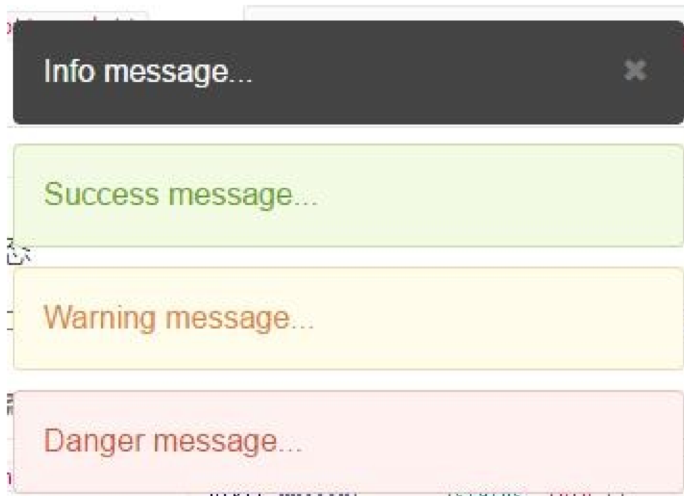
```
UIkit.notify("...", {status:'success'})
```

`warning`

```
UIkit.notify("...", {status:'warning'})
```

`danger`

```
UIkit.notify("...", {status:'danger'})
```



搜索/Search

轻松创建一个好看的搜索框。

用法

搜索组件由搜索表单和搜索文本域组成。注意 使用此组件需要额外添加 `search.css` 文件，在 `css/components` 文件夹中。此组件需要额外添加 `search.js` 文件，在 `js/components` 文件夹中。

Class类	描述
`.uk-search`	添加这个类到 `<form>` 元素中，定义搜索组件。
`.uk-search-field`	添加这个类到 `<input>` 元素中创建一个搜索文本域。

为了使搜索框所必须的JavaScript能够生效，需要添加 `data-uk-search` 属性。

Example



Markup

```
<form class="uk-search" data-uk-search>
    <input class="uk-search-field" type="search" placeholder="">
</form>
```

JSON 搜索结果

在需要用JSON格式化搜索结果的表单中添加 `{source:'PATH/TO/RESULTS'}` 到data属性中，并设置JSON文件的路径 (示例).你可以用 下拉菜单组件 中的下拉菜单来显示搜索结果。搜索的结果会注入并显示在下拉菜单中。你甚至可以用键盘上的上下键来操作这个下拉菜单。

重要 这样的搜索框需要用到 自动完成组件。请确定你已经将他们引入了。

Example



Markup

```
<form class="uk-search" data-uk-search="{source:'my-results.json'}'

    <input class="uk-search-field" type="search" placeholder="">

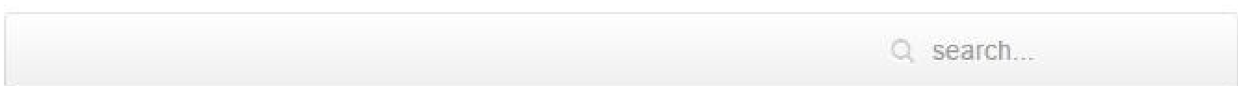
    <!-- 这是通过JavaScript注入了搜索结果的下拉菜单 -->
    <div class="uk-dropdown uk-dropdown-search">
        <ul class="uk-nav uk-nav-search">...</ul>
    </div>

</form>
```

导航条中的搜索

搜索框也可以用在导航条中.

Example



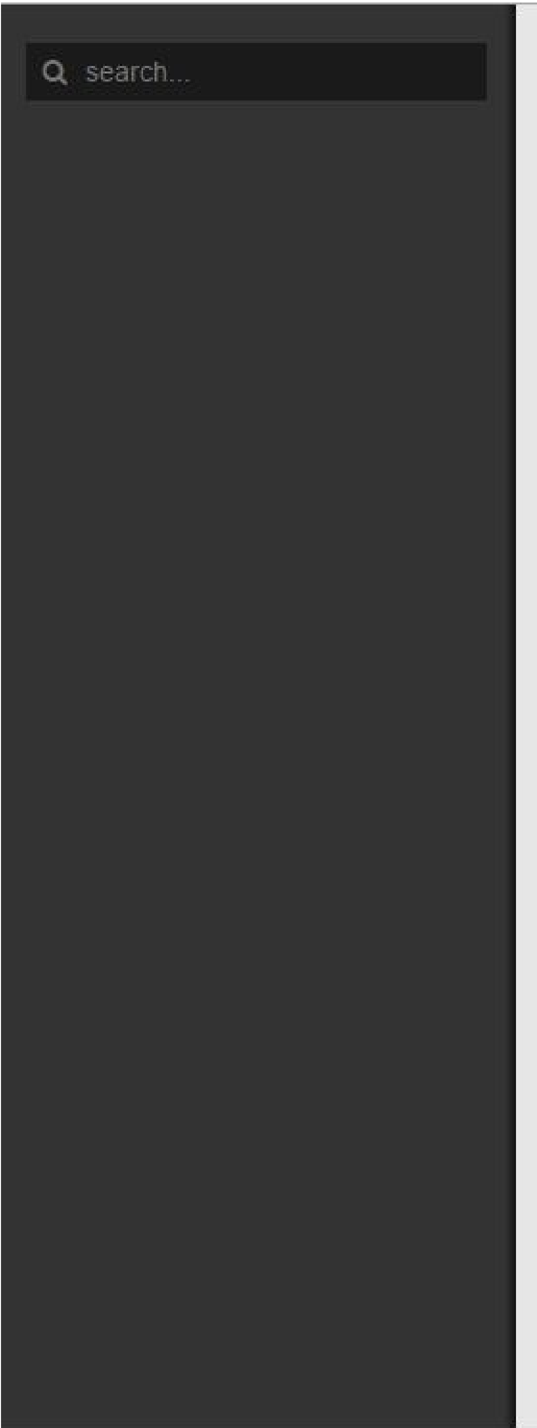
Markup

```
<nav class="uk-navbar">
    <div class="uk-navbar-flip">
        <div class="uk-navbar-content">
            <form class="uk-search" data-uk-search="{source:'my-res
            ...
            </form>
        </div>
    </div>
</nav>
```

抽屉中的搜索

搜索框也可以放在抽屉中。

Example



Markup

```
<!-- 这是开关抽屉边栏的按钮 -->
<button class="uk-button" data-uk-offcanvas="{target:'#my-id'}">...

<!-- 这是抽屉边栏 -->
<div id="my-id" class="uk-offcanvas">
    <div class="uk-offcanvas-bar">
        <form class="uk-search">
        ...
        </form>
    </div>
</div>
```

JavaScript 选项

选项	可用值	默认值	描述
`source`	string	''	数据源的URL
`minLength`	integer	3	触发自动完成的最小输入长度
`param`	string	search	发送Ajax请求的查询名称
`delay`	integer	300	停止输入后的延时

手动初始化

```
var search = UIkit.search(element, { /* options */ });
```

事件

跟自动完成组件的事件一样。

可嵌套/Nestable

创建可以通过拖拽排序的可嵌套式列表。

可嵌套组件允许你通过拖拽排序条目。这是非常有用的，比如在管理界面中希望组织不同的分类或者菜单条目时。

用法

可嵌套列表由列表本身、它的内容条目和可嵌套的面板组成。注意 使用此组件需要额外添加 `nestable.css` 文件，在 `css/components` 文件夹中。此组件需要额外添加 `nestable.js` 文件，在 `js/components` 文件夹中。

class和data属性	描述
`.uk-nestable`	添加此类名到 元素，定义可嵌套组件。
`.uk-nestable-item`	添加此类名到列表的每一个 元素。
`.uk-nestable-panel`	添加此类名到 元素内部的 <div> 元素，给条目设定样式。

NOTE 为了使必要的JavaScript生效，需要添加 `data-uk-nestable` 属性。

Example



Markup

```
<ul class="uk-nestable" data-uk-nestable>
    <li class="uk-nestable-item">
        <div class="uk-nestable-panel"> ... </div>
    </li>
</ul>
```

可嵌套组件的手柄/Nestable handle

默认地，整个可嵌套元素都可以用来拖拽排序。要创建一个替换默认效果的手柄，需要添加 `{handleClass:'uk-nestable-handle'}` 选项到data 属性中。并为你想要用作手柄的元素添加手柄的类名。

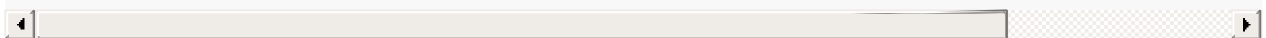
Example



NOTE 在这个例子中，我们使用了 图标组件 中的 `.uk-icon-bars` 类名，来为手柄定义样式。

Markup

```
<ul class="uk-nestable" data-uk-nestable="{handleClass:'uk-nestable
    <li class="uk-nestable-item">
        <div class="uk-nestable-panel">
            <div class="uk-nestable-handle"></div>
            ...
        </div>
    </li>
</ul>
```



可嵌套拨动/Nestable toggle

默认地，整个可嵌套元素都能拖拽排序。要创建一个替换默认效果的手柄，需要添加 `.uk-nestable-toggle` 类名和 `data-nestable-action="toggle"` 属性到可嵌套面板内的 `<div>` 元素。

Example



Markup

```
<ul class="uk-nestable" data-uk-nestable">
    <li class="uk-nestable-item">
        <div class="uk-nestable-panel">
            <div class="uk-nestable-toggle" data-nestable-action="t
            ...
        </div>
    </li>
</ul>
```

多个列表的排序

要实现垮列表的排序，你可以为每个列表添加e `data-uk-nestable="{group:'GROUP-NAME'}"` 属性将他们归为一组。

Example



Markup

```
<ul class="uk-nestable" data-uk-nestable="{group:'my-group'}">...</
<ul class="uk-nestable" data-uk-nestable="{group:'my-group'}">...</
```

禁用嵌套

禁用列表条目的嵌套，你只需添加 `data-uk-nestable="{maxDepth:1}"` 属性。当然你还可以使用此属性来确定要嵌套的深度能有多大。



Markup

```
<ul class="uk-nestable" data-uk-nestable="{maxDepth:1}">...</ul>
```

JavaScript 选项

这是一个如何使用data属性来设置选项的例子:

```
data-uk-nestable="{maxDepth:0, group:'widgets'}"
```

选项	可用值	默认值	描述
`group`	string	false	列表组
`maxDepth`	integer	10	最大嵌套层数
`threshold`	integer	20	以px为单位，开始拖拽的阈值。
	string	ul	列表节点名称

	string	ul	列表节点名称
`itemNodeName`	string	li	条目节点名称
`listBaseClass`	string	uk-nestable	列表的基本类名
`listClass`	string	uk-nestable-list	列表的类名
`listitemClass`	string	uk-nestable-list-item	列表条目的类名
`itemClass`	string	uk-nestable-item	条目类名
`dragClass`	string	uk-nestable-list-dragged	添加到被拖拽列表的类名
`movingClass`	string	uk-nestable-moving	拖动时添加到 `<html>` 元素的类名
`handleClass`	string	uk-nestable-handle	拖拽手柄的类名
`collapsedClass`	string	uk-nestable-collapsed	被收缩的条目的类名
`placeClass`	string	uk-nestable-placeholder	当前被拖拽元素的占位符类名
`noDragClass`	string	uk-nestable-nodrag	带有此类名的元素不会触发拖拽。Elements with this class will not trigger dragging. Useful when having the complete item draggable and not just the handle.
`noChildrenClass`	string	uk-nestable-nochildren	带有此class的元素不再有子级元素。这对于最低层级的条目很有用。
`emptyClass`	string	uk-nestable-	空列表的类名

		empty	

手动初始化

```
var nestable = UIkit.nestable(element, { /* options */ });
```

Events

名称	参数	描述
`start.uk.nestable`	event, nestable object	拖拽开始时触发
`move.uk.nestable`	event, nestable object	移动可嵌套条目时触发
`stop.uk.nestable`	event, nestable object	拖拽终止时触发
`change.uk.nestable`	event, nestable item, action	改变可嵌套条目时触发

可排序/Sortable

创建可排序的网格和列表重新来排列元素的顺序。

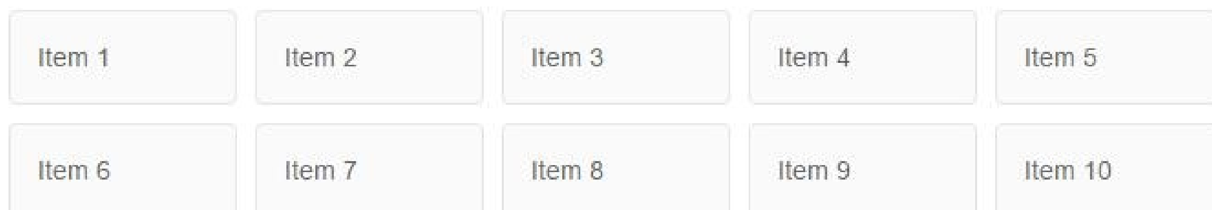
拖拽元素到另一个可排序的网格中的某处，该网格中其它条目会自动适应排列顺序。这将会在诸如排列画廊条目或者菜单条目时显得尤为有用。

用法

要使用这个组件，需要在容器中添加 `.uk-sortable` 类，然后创建子元素来定义这个组件。为了使必要的JavaScript生效，还需要添加 `data-uk-sortable` 属性。注意 使用此组件需要额外添加 `sortable.css` 文件，在 `css/components` 文件夹中。此组件需要额外添加 `sortable.js` 文件，在 `js/components` 文件夹中。

Example

在这里的例子中，我们使用到了 网格组件来放置可排序的条目。



Markup

```
<ul class="uk-sortable" data-uk-sortable>
    <li>...</li>
    <li>...</li>
</ul>
```

任意元素的排序

可排序组件并不限制只能用于 `<ul>` 元素。你可以使用任意块元素作为容器。

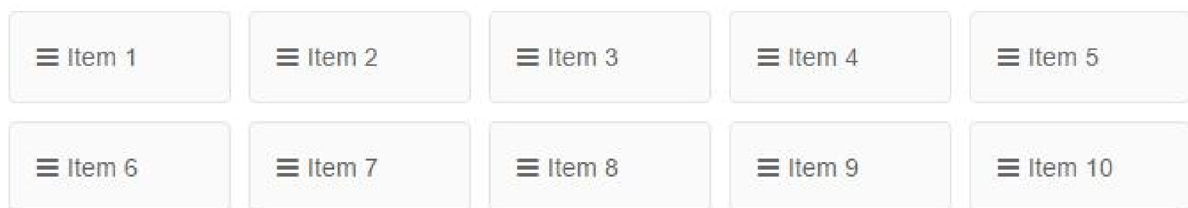
Markup

```
<div class="uk-sortable" data-uk-sortable>
    <div>...</div>
    <div>...</div>
</div>
```

可排序手柄

默认地，整个可排序元素都可以拖拽进行排序。为了创建一个操作手柄，只需为希望作为手柄的元素添加 `{handleClass:'uk-sortable-handle'}` 选项到 data 属性，并添加手柄的class 类名。

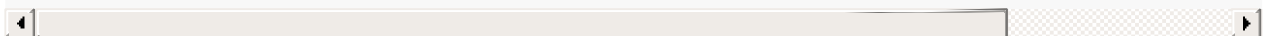
Example



NOTE 在这个例子中，使用了 图标组件 中的 `.uk-icon-bars` 类名来设定手柄的样式。

Markup

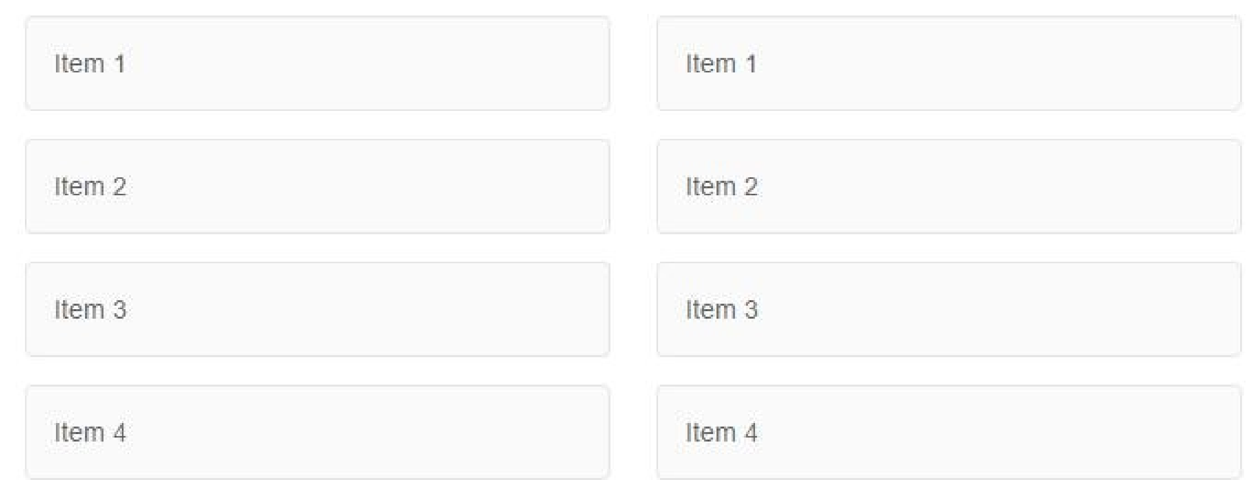
```
<ul class="uk-sortable" data-uk-sortable="{handleClass:'uk-sortable
    <li><div class="uk-sortable-handle"></div>...</li>
</ul>
```



多个列表之间的排序

为了是跨列表的拖拽排序成为可能，你需要为每个列表添加 `data-uk-sortable="{group:'GROUP-NAME'}"` 属性，将它们归为一组。

Example



Markup

```
<ul class="uk-sortable" data-uk-sortable="{group:'my-group'}">...</
<ul class="uk-sortable" data-uk-sortable="{group:'my-group'}">...</
```

JavaScript 选项

这是一个关于如何通过data属性设置选项的例子：

```
data-uk-sortable="{animation:0, dragCustomClass:'dragging'}"
```

选项	可用值	默认值	描述
`group`	string	false	列表的组
`animation`	integer	150	毫秒计时的动画
`threshold`	integer	10	触发元素拖拽的鼠标移动像素距离的阈值
`handleClass`	string	"	自定义类名，用于定义哪些元素可以触发排序
`dragCustomClass`	string	"	添加到被拖拽元素中的自定义类

手动地初始化元素

```
var sortable = UIkit.sortable(element, { /* options */ });
```

Events

Name	Parameter	Description
`start.uk.sortable`	event, sortable object, dragged element	可排序拖拽开始时触发
`move.uk.sortable`	event, sortable object	移动可排序条目时触发
`stop.uk.sortable`	event, sortable object, dragged element	拖拽终止时触发
`change.uk.sortable`	event, sortable object, dragged element, action	改变可排序条目时触发

附着/Sticky

让页面元素保持在视口顶部，比如跟随滚动的导航栏。

用法

创建一个能在页面滚动时能保持在视口顶部的页面元素，添加 `data-uk-sticky` 属性到该元素即可。注意 使用此组件需要额外添加 `sticky.css` 文件，在 `css/components` 文件夹中。此组件需要额外添加 `sticky.js` 文件，在 `js/components` 文件夹中。

Example

附着在顶部

Markup

```
<div data-uk-sticky>...</div>
```

赋予偏移量

还可以定位元素处于视口边缘下的位置。比如，添加 `data-uk-sticky="{top:100}"` 属性创建100px的margin。

Example

在距离顶部边缘100px的位置

Markup

```
<div data-uk-sticky="{top:100}">...</div>
```

添加延迟

为元素添加延迟，这样使它能在页面滚动特定距离后才变成粘连状态，你需要添加一个负偏移值到data属性，比如 `data-uk-sticky="{top:-200}"` 。还可以添加动画让元素可以平滑地再次出现。

Example

滚动100px后再粘连在顶部

Markup

```
<div data-uk-sticky="{top:-200, animation: 'uk-animation-slide-top
```

响应式行为

还可以通过在data属性中添加断点选项，来实现在不同设备上禁用粘连行为，比如 `data-uk-sticky="{media: 640}"` 。另外，还可以使用媒体查询来控制。

Markup

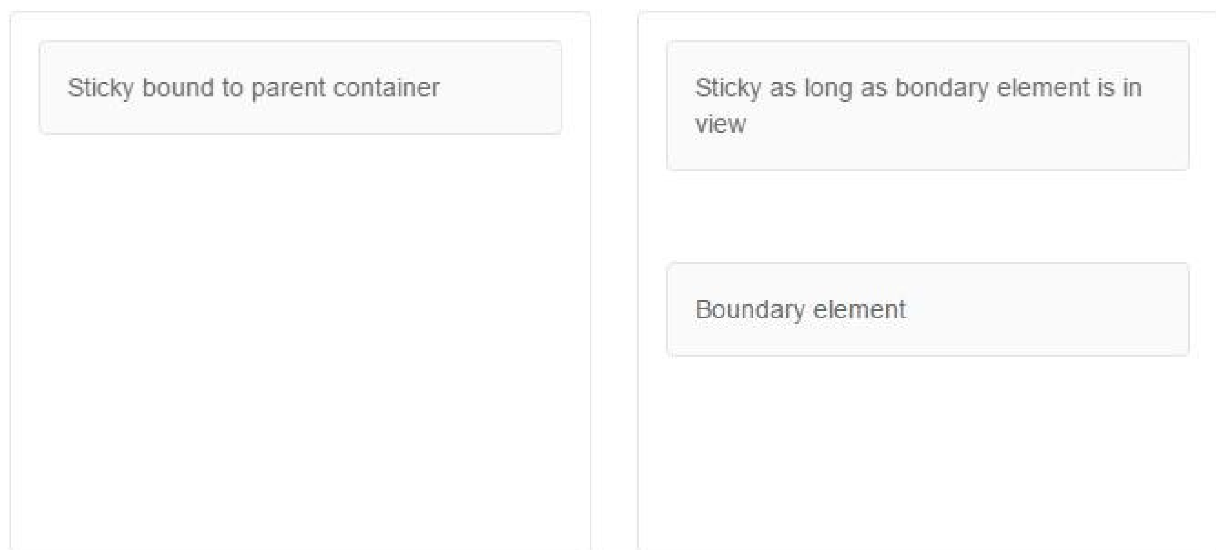
```
<!-- This is basically a shortcode to define a min-width -->
                              <div data-uk-sticky="{media: 640}">

                              <!-- This is a media string using t
                              <div data-uk-sticky="{media: '(min-
```

附着的边界

可以通过设置 boundary 参数定义元素附着行为的边界，使其只在该范围内跟随滚动。

Example



Markup

```
<!-- Bind sticky to its parent element -->
                              <div data-uk-sticky="{boundary: tru

                              <!-- Bind sticky to any element -->
                              <div data-uk-sticky="{boundary: '#m
```

JavaScript 选项

选项	可用值	默认值	描述
`top`	integer	0	触发附着行为的顶部偏移量
`animation`	string	''	UIkit 的动画 class
`clsinit`	string	uk-sticky-init	元素首次附着时进行初始化的class
`clsactive`	string	uk-active	元素附着时添加的 class
`clsinactive`	string	''	元素未附着时添加的 class
`getWidthFrom`	string	''	粘连模式下获取宽度的CSS选择器。默认情况下它从已创建的外层元素获取宽度值。
`media`	integer / string	false	激活状态所需的整型宽度条件，或CSS媒体查询
`target`	boolean	false	确保粘连元素不会在DOM就绪后通过位置散列（location hash）越过目标元素。
`showup`	boolean	false	是否仅在滚动时显示附着的元素
`boundary`	mixed	false	设置为 `true` 将粘连绑定到父元素或使用CSS选择器将粘连绑定到特定元素。

手动初始化元素

```
var sticky = UIkit.sticky(element, { /* options */ });
```

事件

名称	参数	描述
`active.uk.sticky`	event	获得附着效果
`inactive.uk.sticky`	event	脱离附着状态

时间选择器

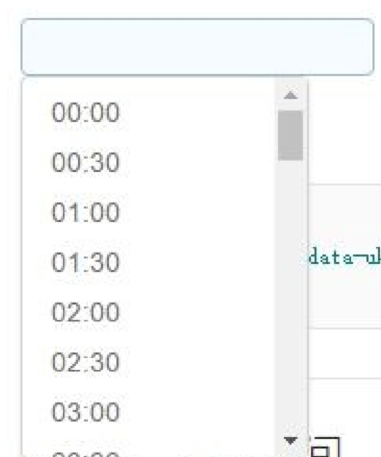
轻松创建可以从预填充下拉菜单中选择时间的时间选择器。

用法

将 `data-uk-timepicker` 属性添加到 `<input>` 元素中就能创建时间选择器。获得焦点后，时间选择器会自动显示预填充的下拉菜单，可以通过键盘上的上下按钮或滚动鼠标进行浏览。注意 使用此组件需要额外添加 `timepicker.js` 文件，在 `js/components` 文件夹中。

重要 时间选择器组件需要自动完成组件才能生效。请确保已经将两者引入页面。

Example



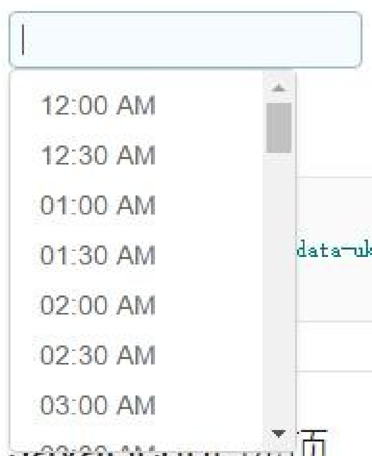
Markup

```
<form class="uk-form">
    <input type="text" data-uk-timepicker>
</form>
```

显示子午线时间

添加*format*选项并设置为 `12h` 即可显示子午线时间。

Example



Markup

```
<form class="uk-form">
    <input type="text" data-uk-timepicker="{format:'12h'}">
</form>
```

JavaScript 选项

这是如果通过属性设置选项的例子：

```
data-uk-timepicker="{format:'12h'}"
```

选项	可用值	默认值	描述
`format`	'24h' or '12h'	'24h'	定义时间表示法
`start`	Integer between 0 and 24	0	起始时间
`end`	Integer between 0 and 24	24	终止时间

手动初始化元素

```
var timepicker = UIkit.timepicker(element, { /* options */ });
```

工具提示/Tooltip

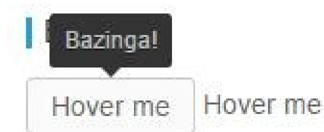
轻松创建一个漂亮的工具提示。

用法

要应用这个组件，需要为某个元素添加 `data-uk-tooltip` 属性。你还需要添加一个 title 属性，它的值即是提示文本。

注意 使用此组件需要额外添加 `tooltip.css` 文件，在 `css/components` 文件夹中。此组件需要额外添加 `tooltip.js` 文件，在 `js/components` 文件夹中。

Example



Markup

```
<button class="uk-button" data-uk-tooltip title="">...</button>
<span data-uk-tooltip title="">...</span>
```

对齐

添加下列选项之一到 `data-uk-tooltip` 属性中来调整提示组件的文本对齐方式。

属性	描述	示例
`pos:'top'`	将提示组件对齐到顶部。	Bazinga! Hover me
`pos:'top-left'`	将提示组件对齐到左上。	Bazinga! Hover me
`pos:'top-right'`	将提示组件对齐到右上。	Bazinga! Hover me
`pos:'bottom'`	将提示组件对齐到底部。	Hover me Bazinga!
`pos:'bottom-left'`	将提示组件对齐到左下。	Hover me Bazinga!
`pos:'bottom-right'`	将提示组件对齐到右下。	Hover me Bazinga!
`pos:'left'`	将提示组件对齐到左侧。	Bazinga! Hover me
`pos:'right'`	将提示组件对齐到右侧。	Hover me Bazinga!

Markup

```
<button class="uk-button" data-uk-tooltip="{pos:'bottom-left'}" ti
```

JavaScript 选项

这是一个如何通过属性来设置选项的示例:

```
data-uk-tooltip="{pos:'bottom-left'}"
```

选项	可用的值	默认值	描述
`offset`	integer	5	与源元素之间的偏移量
`pos`	string	'top'	工具提示组件定位
`animation`	boolean	false	工具提示的淡入动画
`delay`	integer	0	提示组件延迟显示多少毫秒
`cls`	string	''	显示提示时，添加的自定义class
`activeClass`	string	'uk-active'	被拨动的选中类名/Toggled active class

上传

允许用户通过文件表单或占位符域来上传文件

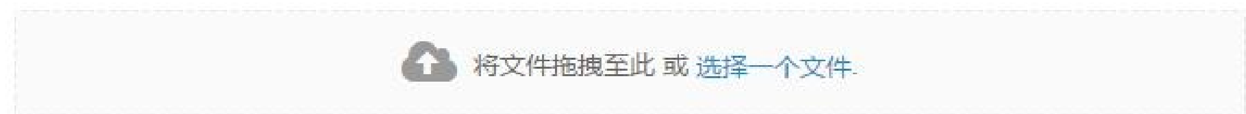
用法

这个Javascript组件运用了最新的 XMLHttpRequest Level 2 规范，提供了通过包含上传进度条的Ajax进行文件上传追踪的功能。本组件提供了两种上传文件的方式：`select` 和 `drop` 。 `select` 请求只能被用在 `<input type="file">` 元素中，而 `drop` 基本可以用在任何元素，通过从桌面将文件拖拽到指定元素就能轻松实现上传。记住，本组件并不在服务器上处理文件上传。

注意 使用此组件需要额外添加 `upload.css` 文件，在 `css/components` 文件夹中。此组件需要额外添加 `upload.js` 文件，在 `js/components` 文件夹中。

上传组件需要根据你的要求单独进行实施。在我们的例子中，我们使用占位符和文件表单，同时使用了 `drop` 和 `select` 请求。另外，还是用了进度条来显示上传进度。

Example



Markup

```
<div id="upload-drop" class="uk-placeholder">
    Info text... <a class="uk-form-file">Select a file<input id="u
</div>

<div id="progressbar" class="uk-progress uk-hidden">
    <div class="uk-progress-bar" style="width: 0%;">...</div>
</div>
```

JavaScript

为了创建 `select` 和 `drop` 上传监听器，你需要使用目标元素和选项来实例化每个上传class，以定义回调和其他有用的设置。

```
<script>

    $(function(){

        var progressbar = $("#progressbar"),
            bar         = progressbar.find('.uk-progress-bar'),
            settings    = {

            action: '/', // 上传路径 url

            allow : '*.(jpg|jpeg|gif|png)', // 只允许上传图片

            loadstart: function() {
                bar.css("width", "0%").text("0%");
                progressbar.removeClass("uk-hidden");
            },

            progress: function(percent) {
                percent = Math.ceil(percent);
                bar.css("width", percent+"%").text(percent+"%");
            },

            allcomplete: function(response) {

                bar.css("width", "100%").text("100%");

                setTimeout(function(){
                    progressbar.addClass("uk-hidden");
                }, 250);

                alert("Upload Completed")
            }
        };

        var select = UIkit.uploadSelect($("#upload-select"), settir
            drop   = UIkit.uploadDrop($("#upload-drop"), settings),
    });

</script>
```

JavaScript 选项

选项	可用值	默认值	描述
`action`	string	''	上传的目标URL
`single`	boolean	true	逐一进行文件发送
`param`	string	files[]	传递查询名称
`params`	JSON Object	{}	额外的请求参数
`allow`	string	.	文件过滤器
`filelimit`	integer	false	文件上传数量限制
`type`	(text | json)	text	来自服务器的响应类型

回调事件

名称	参数
`before`	settings, files
`beforeAll`	files
`beforeSend`	xhr
`progress`	percent
`complete`	response, xhr
`allcomplete`	response, xhr
`notallowed`	file, settings
`loadstart`	event
`load`	event
`loadend`	event
`error`	event
`abort`	event
`readystatechange`	event